

Test 1 Review

CSCI 432

Test 1 Logistics

1. During class on Friday 9/25.
2. You can bring your book and any notes you would like, but no electronic devices.
3. You may assume anything proven in class or on homework.
4. Three questions (12 points):
 - 1) Prove a greedy algorithm is optimal (5 points).
 - 2) Prove performance of a randomized algorithm (5 points).
 - 3) Prove or disprove a claim (2 points).

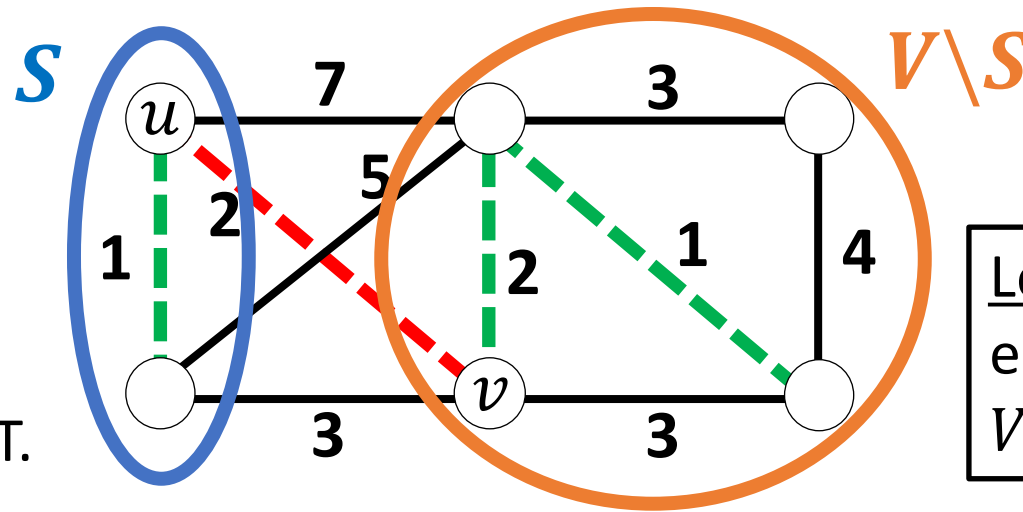
Kruskal's MST Algorithm

Algorithm: Add the edge with smallest weight, that does not create a cycle.

Proof of optimality: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Kruskal's algorithm.

Consider the iteration that some edge $e = (u, v)$ is added by Kruskal's algorithm. Let S be the set of u and all nodes already connected to u . Clearly $u \in S$ and $v \in V \setminus S$ (otherwise adding e would have created a cycle). We are picking the cheapest edge that crosses the cut (otherwise the cheaper edge would have been selected since it would not have created a cycle either). By the cut property lemma, this edge must be part of the MST.

Thus, every edge found by Kruskal's algorithm is part of the MST, and since the edges found form a spanning tree, it is the MST.



Lemma: The cheapest edge between $S \subseteq V$ and $V \setminus S$ is part of the MST.

Prim's MST Algorithm

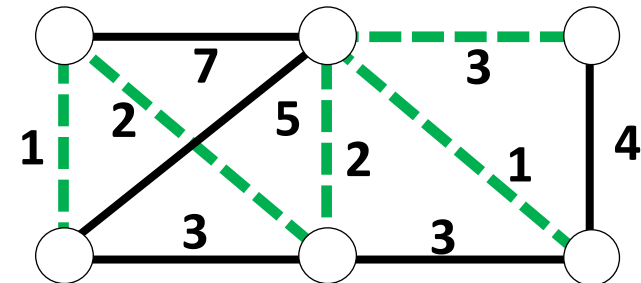
Algorithm: Beginning at any node, add the node that can be connected as cheaply as possible to the tree we are building.

Proof of validity & optimality: Let $G = (V, E)$, and $T \subseteq E$ be the set of edges resulting from Prim's algorithm.

T is a tree because it consists of a single connected component and we never add an additional edge to an already connected node (i.e. no cycles).

T is a spanning tree because we added nodes (edges) until there were no nodes left to add.

T is an MST because at each step we add the cheapest edge between the current tree and the remaining nodes (cut property).



Single Room Scheduling

Greedy decision: Select the next course with the earliest compatible finish time.

Proof of optimality: Let $\mathcal{C}$ be the set of courses, $S_{ALG} \subseteq \mathcal{C}$ be the greedy algorithm's selection, and $S_{OPT} \subseteq \mathcal{C}$ be an optimal selection, all sorted by increasing finish time.

Suppose $S_{ALG}[i] = S_{OPT}[i]$, for all $i < k$ and $S_{ALG}[k] = c_i \neq c_j = S_{OPT}[k]$.

Create the revised schedule $S'_{OPT} = S_{OPT} \setminus \{c_j\} \cup \{c_i\}$. (I.e., Swap $S_{ALG}[k]$ for $S_{OPT}[k]$)

c_i is compatible with previous courses in S'_{OPT} since $S_{ALG}[i] = S_{OPT}[i] = S'_{OPT}[i]$, for all $i < k$

c_i is compatible with subsequent courses in S'_{OPT} since $f_i \leq f_j$. Otherwise, the greedy algorithm would have selected c_j instead of c_i .

So S'_{OPT} is a valid schedule with the same number of courses as S_{OPT} , so S'_{OPT} is also optimal.

We can then proceed inductively and show that each course in S_{OPT} can be replaced by the corresponding course in S_{ALG} without violating compatibility. Since replacing every course in S_{OPT} with the courses in S_{ALG} keeps the solution optimal, S_{ALG} must be optimal. (i.e., we translated S_{OPT} into S_{ALG} at no extra cost).

Optimal?

```
room_minimization(courses c)
```

```
    F = B = S =  $\emptyset$ 
```

```
    room_num = 0
```

```
    foreach timeslot t
```

```
        foreach c in C
```

```
            if c.finish == t
```

```
                F.add(B.getBookedRoom(S.getroom(c)))
```

```
    foreach c in C
```

```
        if c.start == t
```

```
            if F.isEmpty()
```

```
                room_num += 1
```

```
                F.add(room_num)
```

```
            room = F.getFreeRoom()
```

```
            S.schedule(c, room)
```

```
            B.add(room)
```

```
    return S
```

What is the smallest possible number of rooms?

The number of concurrent courses.

If we identify a lower bound intrinsic to the problem (i.e., can't do is with fewer than X rooms), and our algorithm does it with X rooms, our algorithm is optimal.

rooms needed $\geq$ # concurrent courses.
New room added $\Leftrightarrow$ all other rooms in use
(i.e., # rooms used = max[# concurrent courses])
So, it's optimal.

$B = \{$

$S = \{c_1 \rightarrow 1, c_3 \rightarrow 2, c_2 \rightarrow 1, \dots\}$

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

Can you set the variables to **true** or **false** so that ϕ evaluates to **true**?

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$x_1 = \textit{false}$$

$$x_2 = \textit{true}$$

SAT & 3-SAT

$$\begin{array}{ccccc}\phi = & (x_1 \vee x_1 \vee x_2) & \wedge & (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) & \wedge & (\overline{x_1} \vee x_2 \vee x_2) \\ & \downarrow & & \downarrow & & \downarrow \\ & (F \vee F \vee T) & & (T \vee F \vee F) & & (T \vee T \vee T)\end{array}$$

$$x_1 = \textit{false}$$

$$x_2 = \textit{true}$$

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$\begin{array}{c} \downarrow \\ (F \vee F \vee T) \\ \downarrow \\ T \end{array}$$

$$\begin{array}{c} \downarrow \\ (T \vee F \vee F) \\ \downarrow \\ T \end{array}$$

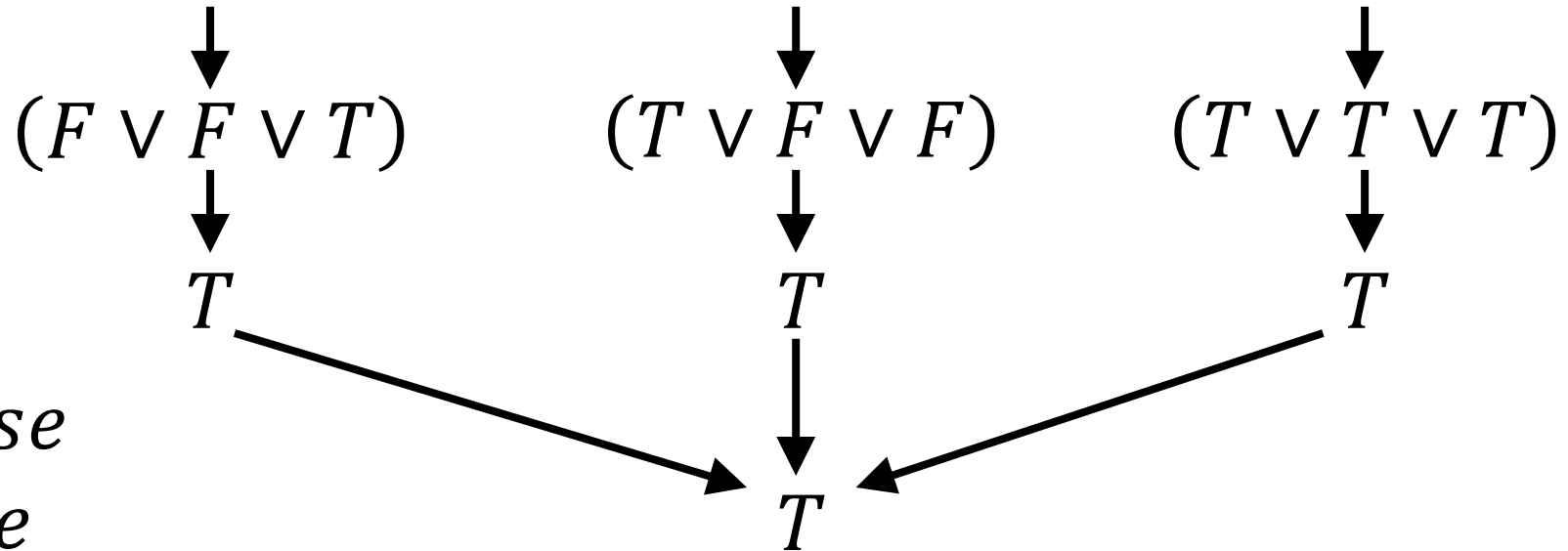
$$\begin{array}{c} \downarrow \\ (T \vee T \vee T) \\ \downarrow \\ T \end{array}$$

$$x_1 = \textit{false}$$

$$x_2 = \textit{true}$$

SAT & 3-SAT

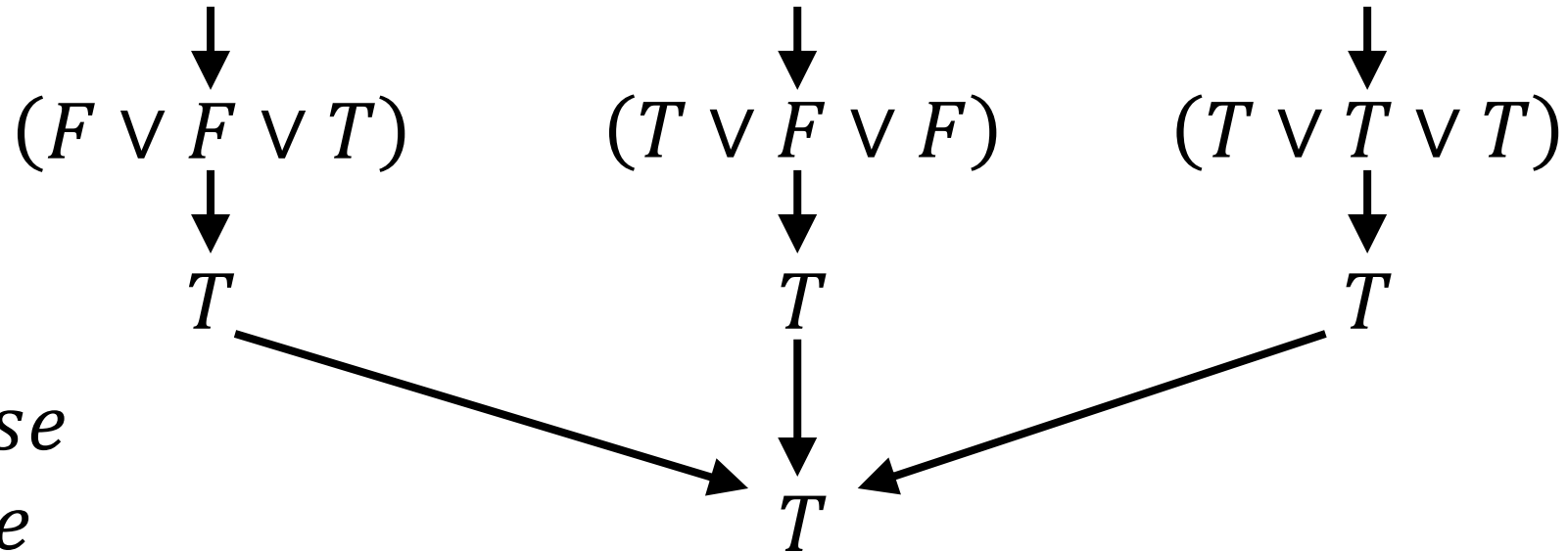
$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$



$x_1 = \text{false}$
 $x_2 = \text{true}$

SAT & 3-SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$



$x_1 = false$
 $x_2 = true$

$SAT = \{\langle \phi \rangle: \phi \text{ is a satisfiable formula}\}$

$3SAT = \{\langle \phi \rangle: \phi \text{ is a satisfiable formula with 3 variables per clause}\}$

Max 3-SAT

Given a 3-SAT instance (with unique variables in each clause), set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

$$\{\text{true}, \text{true}, \text{false}, \text{false}\} \rightarrow \text{true} \wedge \text{false} \wedge \text{true}$$

$$\{\text{true}, \text{false}, \text{true}, \text{false}\} \rightarrow \text{true} \wedge \text{true} \wedge \text{true}$$

Max 3-SAT

Given a 3-SAT instance (with unique variables in each clause), set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} $\rightarrow$ true $\wedge$ false $\wedge$ true

{true, false, true, false} $\rightarrow$ true $\wedge$ true $\wedge$ true

Problem:

Consider the following randomized algorithm for Max 3-SAT: Assign $x_i = \text{true}$ with probability $1/2$. Otherwise, set it to false.

Show that this algorithm is expected to satisfy $\frac{7}{8}$ of the clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3}\right) + 0 \left(\frac{1}{3}\right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} & \\ \text{expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} &= E[ALG] = E[\sum_e X_e] \\ \text{expect to satisfy} &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} & \\ \text{expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied = ?

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied = ?

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3}\right) + 0 \left(\frac{1}{3}\right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} & \\ \text{expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case
Probability of **(F,F,F)**?

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we} &= E[ALG] = E[\sum_e X_e] \\ \text{expect to satisfy} &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case
Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3}\right) + 0 \left(\frac{1}{3}\right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we} &= E[ALG] = E[\sum_e X_e] \\ \text{expect to satisfy} &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we} &= E[ALG] = E[\sum_e X_e] \\ \text{expect to satisfy} &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

$$1 - \frac{1}{8} = \frac{7}{8}$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} & \\ \text{expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = ?$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{\# edges we} & \\ \text{expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1 \left(\frac{7}{8} \right) + 0 \left(\frac{1}{8} \right) = \frac{7}{8}$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1 \left(\frac{7}{8} \right) + 0 \left(\frac{1}{8} \right) = \frac{7}{8}$$

$$\begin{aligned} \# \text{ clauses we expect to satisfy} &= E[ALG] = E[\sum_c Z_c] \\ &= \sum_c E[Z_c] \end{aligned}$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1 \left(\frac{7}{8} \right) + 0 \left(\frac{1}{8} \right) = \frac{7}{8}$$

$$\begin{aligned} \# \text{ clauses we expect to satisfy} &= E[ALG] = E[\sum_c Z_c] \\ &= \sum_c E[Z_c] \\ &= \sum_c \frac{7}{8} = \frac{7}{8} |C| \end{aligned}$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \# \text{ edges we expect to satisfy} &= E[ALG] = E[\sum_e X_e] \\ &= \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Goal: Show it satisfies $\frac{2}{3}$ of edges.

Max 3-SAT

Algorithm: Assign $x_i = \text{true}$ with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1 \left(\frac{7}{8} \right) + 0 \left(\frac{1}{8} \right) = \frac{7}{8}$$

$$\begin{aligned} \# \text{ clauses we expect to satisfy} &= E[ALG] = E[\sum_c Z_c] \\ &= \sum_c E[Z_c] \\ &= \sum_c \frac{7}{8} = \frac{7}{8} |C| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{7}{8} |C|$$

Goal: Show it satisfies $\frac{7}{8}$ of clauses.