CSE 421 Section 9

Min-Cut & NP Intro

Administrivia

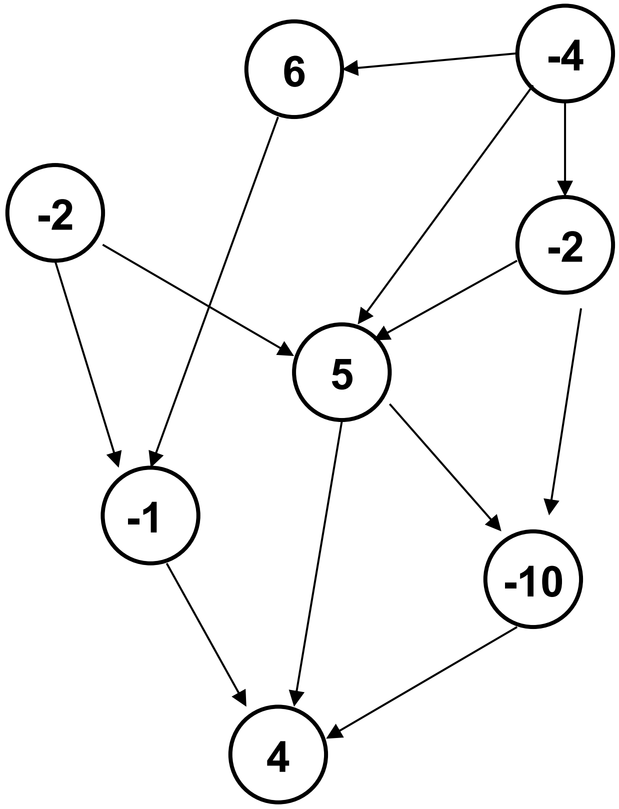
Announcements & Reminders

- HW8
  - Due Friday!

Task Selection

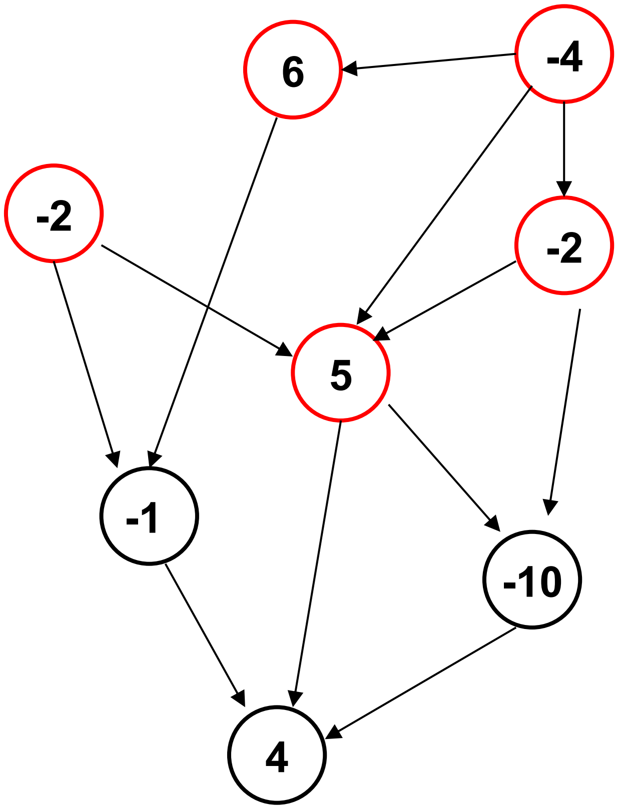
Task Selection

- Precedence graph G=(V,E)
- Each v in V has a profit p(v)
- A set F is *feasible* if when w in F, and (v,w) in E, then v in F.
- Find a feasible set to maximize the profit



Task Selection

- Try to compute by yourself.
- How do we transfer it into Min-Cut problem?

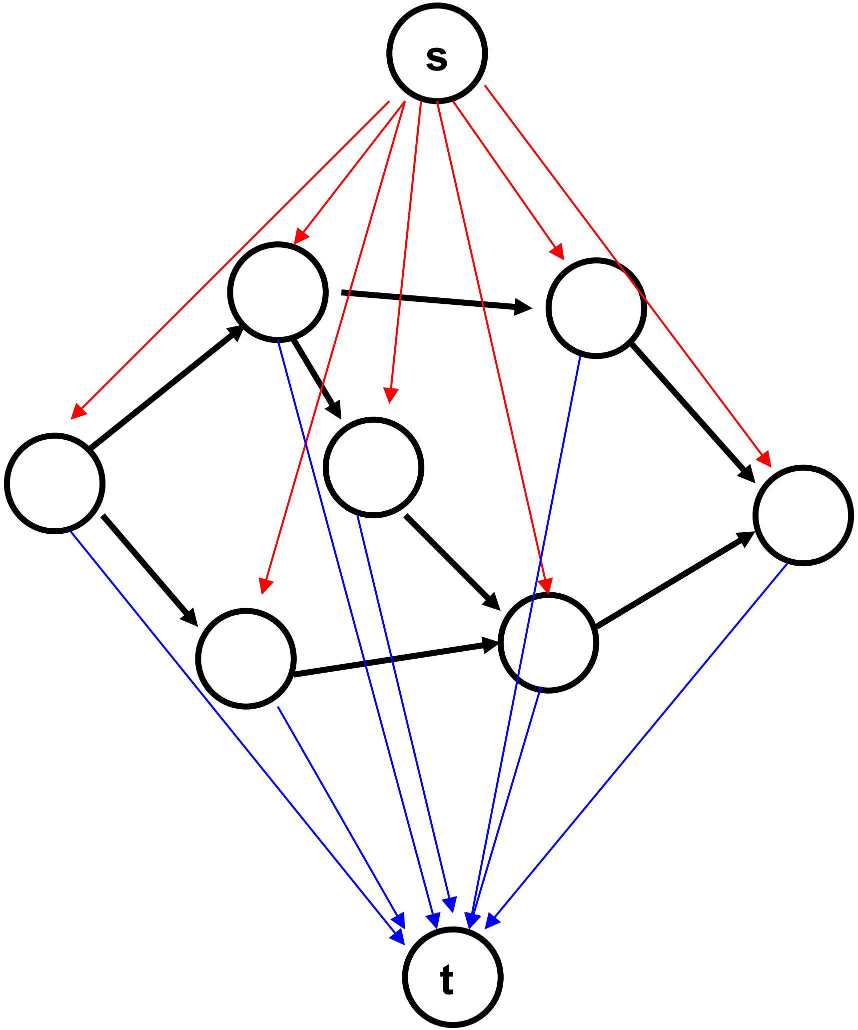

6
-4
-2
-2
5
-1
-10
4

Precedence graph construction

- 1. How to build the edge?
- 2. How to set the edge costs?
- 3. How to make sure it is feasible?

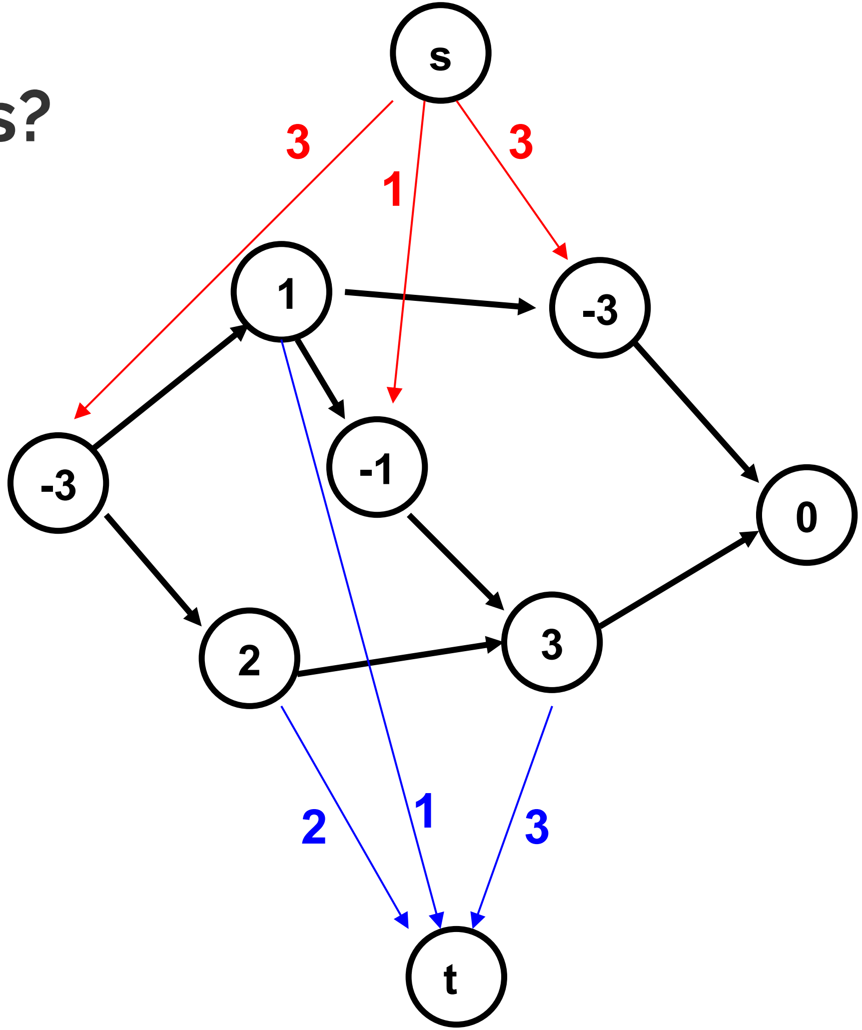
How to build the edge?

- Add vertices s, t
- Each vertex in V is attached to s and t with finite capacity edges



How to set the edge costs?

- If $p(v) > 0$,
  - $cap(v,t) = p(v)$
  - $cap(s,v) = 0$
- If $p(v) < 0$
  - $cap(s,v) = -p(v)$
  - $cap(v,t) = 0$
- If $p(v) = 0$
  - $cap(s,v) = 0$
  - $cap(v,t) = 0$



How to make sure it is feasible?

- Each edge in E has infinite capacity.
- Why?

- The sink side of a finite cut is a feasible set.
- No edges permitted from S to T
- If a vertex is in T, all of its ancestors are in T

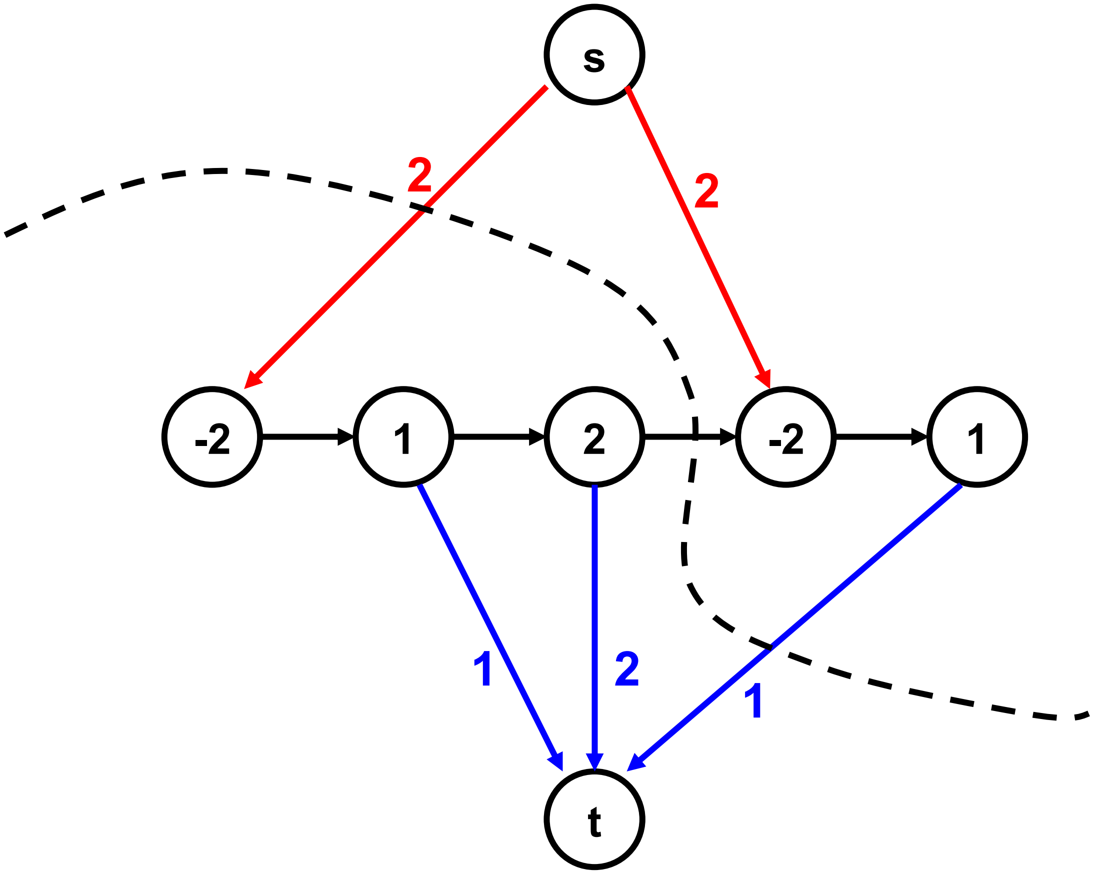
Why Min-Cut gives optimal solution?

- $Cost(W) = \Sigma_{\{w \text{ in } W;\ p(w) < 0\}} -p(w)$
- $Benefit(W) = \Sigma_{\{w \text{ in } W;\ p(w) > 0\}}\ p(w)$
- Profit(W) = Benefit(W) – Cost(W)

- Maximum cost and benefit
  - C = Cost(V)
  - B = Benefit(V)

Express Cap(S,T)

- Cap(S,T) = Cost(T) + Ben(S) = Cost(T) + Ben(S) + Ben(T) – Ben(T)
- = B + Cost(T) – Ben(T) = B – Profit(T)



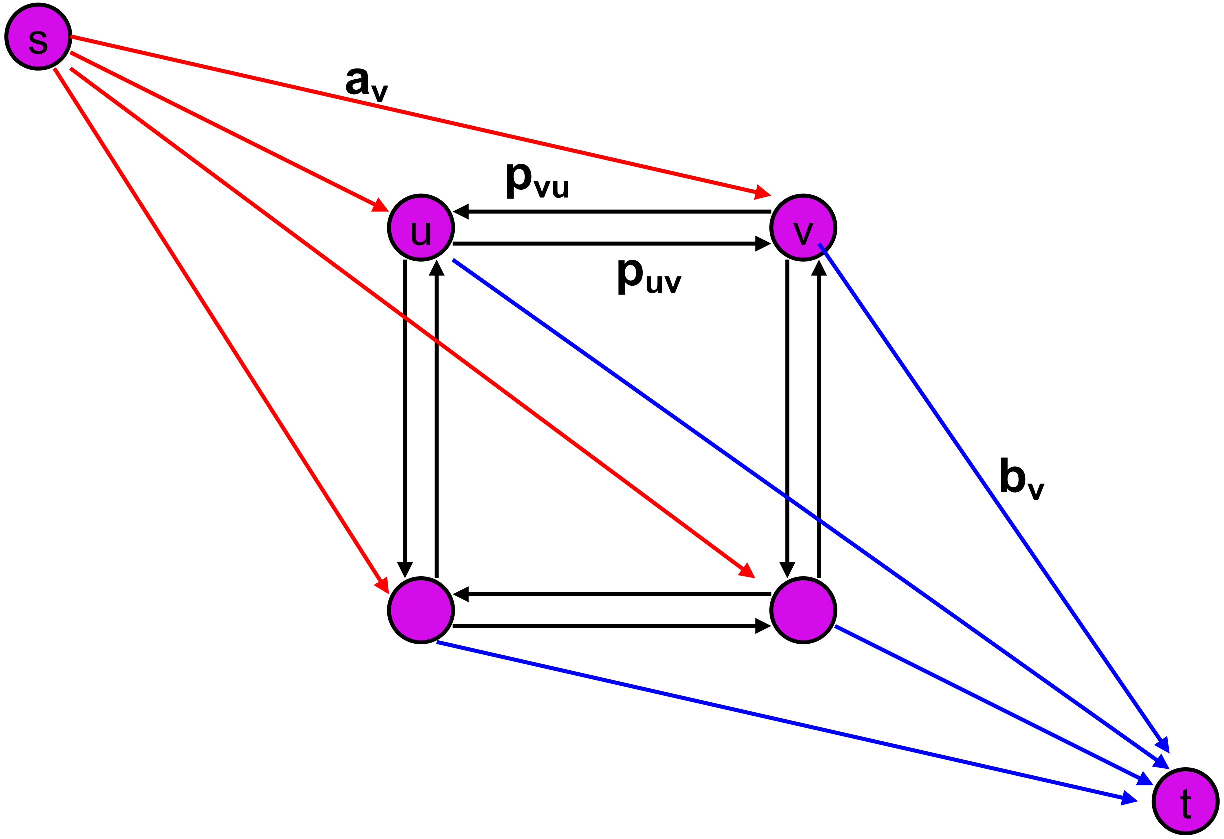
s
2
2
-2
1
2
-2
1
1
2
1
t

Image Segmentation

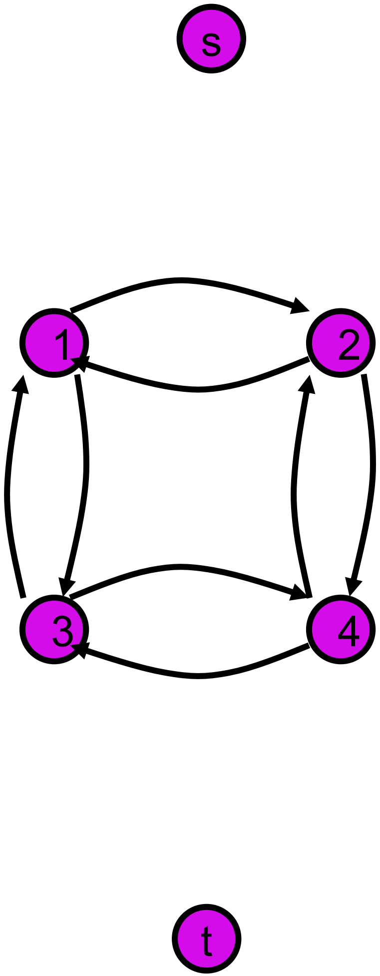
Image analysis

- a_i: value of assigning pixel i to the foreground
- b_i: value of assigning pixel i to the background
- p_{ij}: penalty for assigning i to the foreground, j to the background or vice versa
- A: foreground, B: background
- $Q(A,B) = \Sigma_{\{i \text{ in } A\}} a_i + \Sigma_{\{j \text{ in } B\}} b_j - \Sigma_{\{(i,j) \text{ in } E,\ i \text{ in } A,\ j \text{ in } B\}} p_{ij}$

Mincut Construction



a: [4,9,2,3]
b: [4,7,4,2]
p1:[0,1,1,0]
p2:[2,0,0,3]
p3:[1,0,0,4]
p4:[0,2,1,0]



Best Result

- Choose 2 and 4 for foreground.

P&NP

What is P?

- Decision problems with polynomial time algorithms

What is NP?

- Problems solvable in non-deterministic polynomial time
- Problems where “yes” instances have polynomial time checkable certificates

How to show it is P?

- Is x a multiple of y?
- Is d the minimal distance from S to T?
- Is the edit distance between x and y less than d?
- Is the max profit of feasible set w?

- Division, Mod
- Shortest Path
- Dynamic Programming
- Network Flow

How to show it is NP?

- 3-SAT
- Independent set of size K – The Independent set of size K
- K-coloring a graph – Assignment of colors to the vertices

3-SAT

- SAT: Does a given CNF formula have a satisfying formula?
- 3-SAT: each clause is limited to exactly three literals.
- The literals within a clause can be either a Boolean variable or its negation, and the clauses are connected by logical AND operators.
- Instance S and certificate T:

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

$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$$

How to check the certificate T?

How to check the certificate T?

- Take given values into Boolean expression. Only takes O(n).

Independent set

- A set of vertices in a graph, no two of which are adjacent.

How to check the given independent set?

How to check the given independent set?

- Go through all edges (u,v). Check whether u and v are both in the given independent set. O(m).

K-coloring

- Give all vertices a color (No more than K different colors). No two adjacent vertices are of the same color.

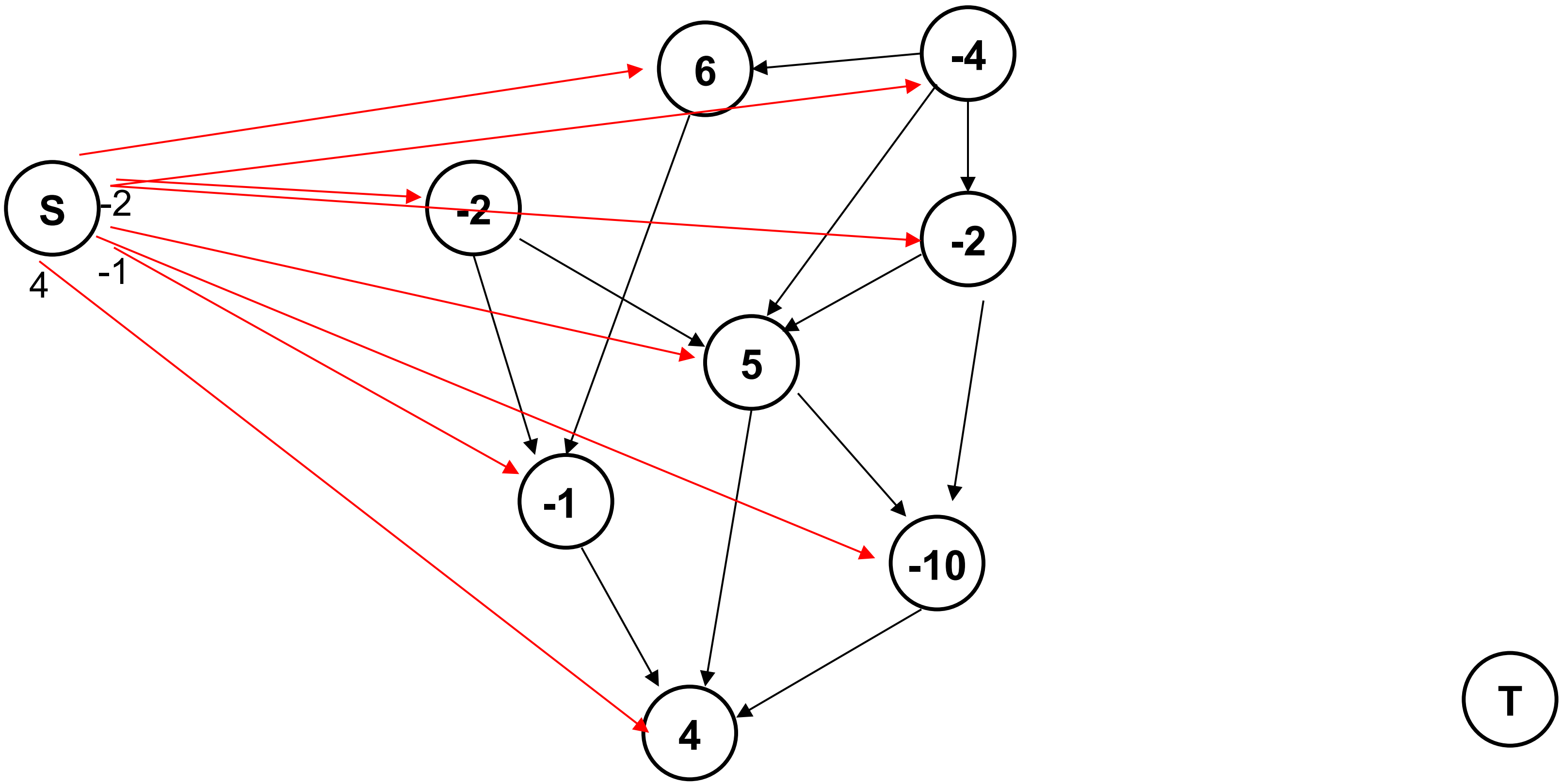
How to check

How to check

- Go through all edges (u,v). Check whether u and v are in the same color. O(m).

That’s All, Folks!

Thanks for coming to section this week!
Any questions?


S
-2
-1
4
6
-4
-2
-2
5
-1
-10
4
T