CS260: ASSIGNMENT 2

Hao Liu(185896) Mohamed Bouaziz(184732) Tongzhou Gu(186085) Juyi Lin(187176)
Jinjie Mai(179794)
{hao.liu, mohamed.bouaziz, tongzhou.gu, juyi.lin, jinjie.mai}@kaust.edu.sa

1 Problem 1

Let G be a connected undirected weighted graph such that no two edges have the same weight. Prove that G has a unique minimum cost spanning tree.

We can prove that Prim algorithm can find a minimum cost spanning tree.

Basis: Obviously holds when there is only one node.

Generalization: If a step holds, the current set of edges is F , which belongs to T the MST, and the next step is to join the edge e .

If e belongs to T , then it holds.

Otherwise consider another edge f on the ring T+e that can be added to the current edge set .

First, the weight of f must not be smaller than the weight of e, otherwise f would be chosen over e. Then, the weight of f must not be smaller than the weight of e.

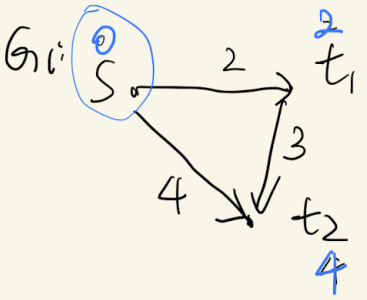
Then, the weight of f must be no greater than the weight of f , otherwise T+e-f would be a smaller spanning tree.

Thus, e and f have equal weights, and T+e-f is also a minimum spanning tree and contains F .

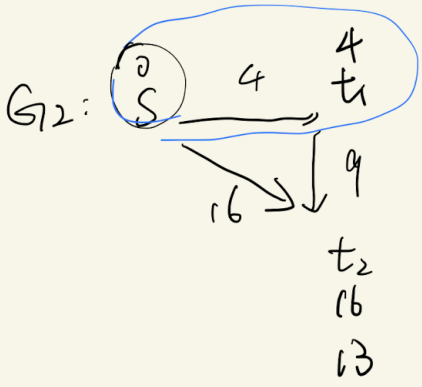
And since no two edges have the same weight, there is no any f in this proces, the minimum spanning tree is unique.

2 Problem 2

2. Disprove.



for G_1,
shortest path from $s \to t_2 = 4$



for G_2.
shortest path from $s \to t_2$ is
$s \to t_1 \to t_2 = 13$

So it is different path.

3 Problem 3

We propose the following algorithm:

1. We keep two sets, the set $\mathcal{A}$ containing all the selected sessions, and the set $\mathcal{U}$ containing all the unselected sessions, which we initialize as all the sessions at the beginning.
2. We select the session with the earliest ending time from $\mathcal{U}$ and put it into $\mathcal{A}$. Next, we delete all the sessions in $\mathcal{U}$ which are conflict with this newly selected session (including itself).
3. Repeat Step 2 until $\mathcal{U}$ is empty, then $\mathcal{A}$ will be our output.

Proof: Suppose $\mathcal{A}$ is not the optimal solution and our optimal solution is $\mathcal{O}$. Let's denote the selected session in $\mathcal{A}$ as i_k and the optimal sessions in $\mathcal{O}$ as j_k.

Lemma : For all $r \leq k$ we have $F_{i_r} \leq F_{j_r}$. We use introduction to prove this.

1. When $r = 1$ it's obvious $F_{i_1} \leq F_{j_1}$.
2. We hypothesis for all $r \leq k$ we have $F_{i_r} \leq F_{j_r}$.
3. Let's prove $F_{i_{k+1}} \leq F_{j_{k+1}}$. With $F_{j_k} \leq S_{j_{k+1}}$ and our hypothesis, we have $F_{i_k} \leq S_{j_{k+1}}$. So we should be able to choose j_{k+1} with our greedy algorithm. Because our greedy algorithm chooses the job with the earliest ending time, we have $F_{i_{k+1}} \leq F_{j_{k+1}}$.
4. Thus, the lemma is proved from the induction.

Let's suppose the selected session in $\mathcal{A}$ as $i_1, i_2, ..., i_k$ and the optimal sessions in $\mathcal{O}$ as $j_1, j_2, ..., j_m$ with m sessions. Suppose $\mathcal{A}$ is not the optimal we have $k < m$. But from our lemma, $F_{i_k} \leq F_{j_k}$. Then there will be a session j_{k+1} that starts after j_k and i_k end but is not chosen by our greedy algorithm. Such a contradiction implies our greedy algorithm is the optimal.

4 Problem 4

PRIM's algorithm to determine the Minimum Spanning Tree on the given graph in Figure 1.

5 Problem 5

5. $k = \frac{10}{2} = 5$ $n = 10$ so the aim is to find 5th largest number

step1: $\lceil 0.9474 \times 10 \rceil = 10$

so we take 19.

$S^- = \{12, 4, 1, 17, 10, 18, 5, 3, 9\}$ 19 $S^+ = \phi$

$|S^-| = 9 > k-1 = 4$ so call Select(S^-, k)

step2: $\lceil 9 \times 0.6753 \rceil = 7$ so we take 5

$S^- = \{4, 1, 3\}$ 5 $S^+ = \{12, 17, 10, 18, 9\}$

$|S^-| = 3 < k-1 = 4$ $k-1-1 = 5-1-3 = 1$

So call select$(S^+, 1)$

step3: $\lceil 0.2297 \rceil \times 5 = 2$ so we take 17

$S^- = \{12, 10, 9\}$ 17 $S^+ = \{18\}$

$|S^-| = 3 > k-1 = 0$ select$(S^-, 1)$

step4: $\lceil 0.2868 \times 3 \rceil = 1$ so we take 12

$S^- = \{10, 9\}$ 12 $S^+ = \phi$

$|S^-| = 2 > k-1 = 0$ so call select$(S^-, 1)$

step5: $\lceil 0.4932 \times 2 \rceil = 1$ so we take 10

$S^- = \{9\}$ 10 $S^+ = \phi$ $|S^-| = 1 > k-1 = 0$

Select$(S^-, 1)$

step6: $\lceil 0.6638 \times 1 \rceil = 1$

$S^- = \phi$ 9 $S^+ = \phi$

$|S^-| = 0 = k-1$

so 9 is the 5st

Answer

6 Problem 6

Apply contraction algorithm to the given graph. Edges should be contracted using the given order of edges such that the edge with the lowest index is contracted first.

Edge numbering: (a,b) – 1, (a,d) – 2, (b,d) – 3, (c,d) – 4, (c,g) – 5, (d,g) – 6, (d,f) – 7, (d,e) – 8, (e,f) – 9, (f,g) – 10.

7 Problem 7

T.

When we choose the 1st 0, the probability is $\frac{k}{n}$, so the expected step is $\frac{n}{k}$.

When choosing the 2st 0, the probability is $\frac{k-1}{n}$. so the expected step is $\frac{n}{k-1}$.

⋮

When we choose the last 0, the probability is $\frac{1}{n}$, so the expected step is n.

So in all the expected steps is

$$n\left(\frac{1}{k}+\frac{1}{k-1}+\cdots+1\right)$$

$\ln k \le 1+\frac{1}{2}+\cdots+\frac{1}{k} \le \ln k+1$. so $n(1+\cdots+\frac{1}{k}) \le n\cdot(\ln k+1)$

$< n(\ln n+1)$

So the expected number of steps is $O(n\log n)$

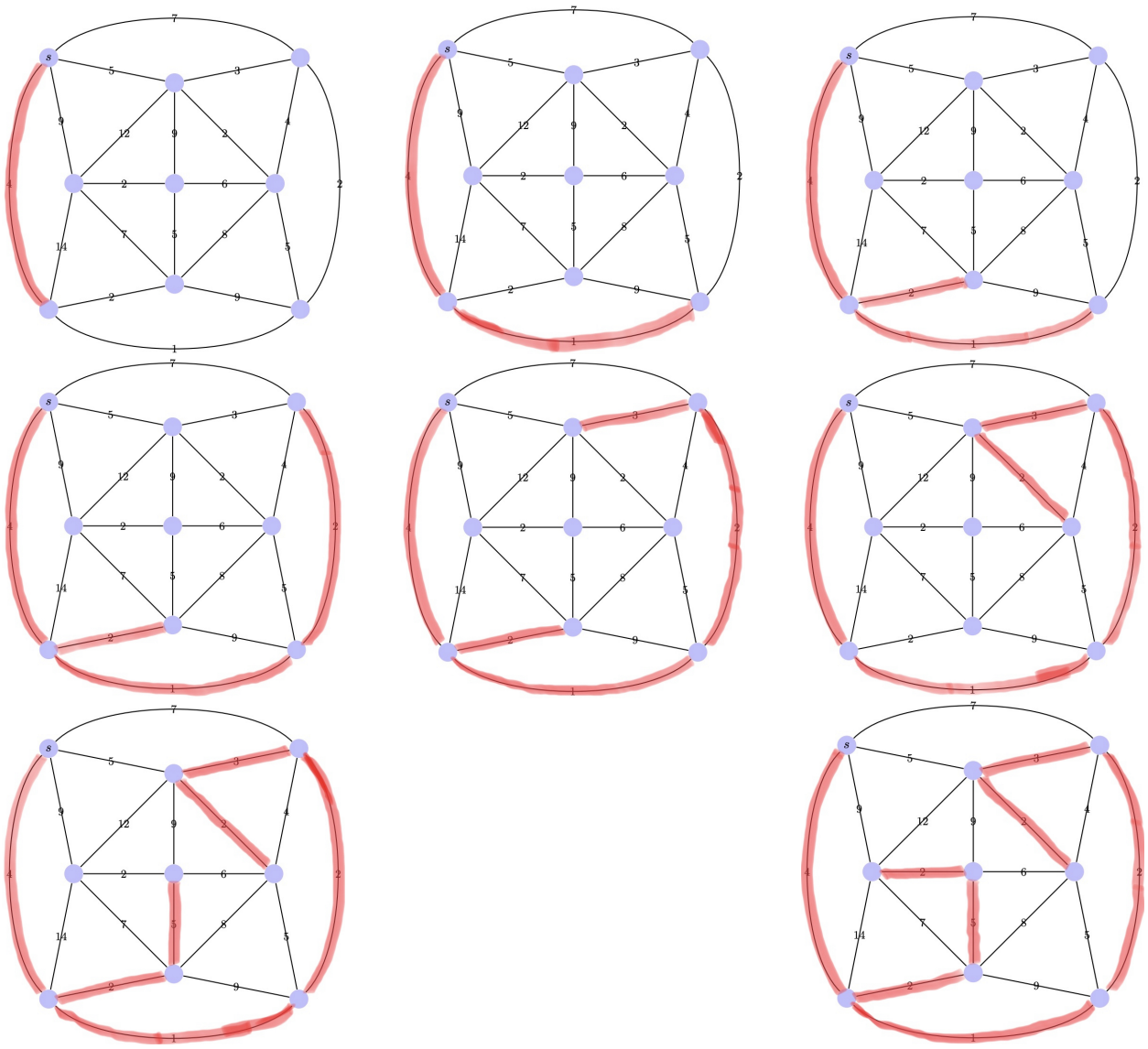


Figure 1: problem 4

8 Problem 8

NP Proof. Let's prove $DS(G, k)$ belongs to NP. The certificate is a subset S of V in size k. The validation algorithm will traverse every vertex in V to check if it's in S or a neighbor of the nodes in S, whose time complexity will be $O(|V| + |E|)$, obviously.

NP-Hard Proof. Let's prove $DS(G, k)$ belongs to NP-Hard. We use the fact that the vertex cover problem is NP-Hard. Suppose the vertex cover of G is a set C in size k, in which every edge in G has at least one end in C. Then this set C will also be our solution for the dominating set problem. Because for each vertex v in C, v is in V, or there's an edge between v in C and another neighbor vertex u not in C but in V. That is, vertex cover and dominating set are both NP-Hard problems.

Therefore, dominating set is NP-Complete as it's NP and NP-Hard.

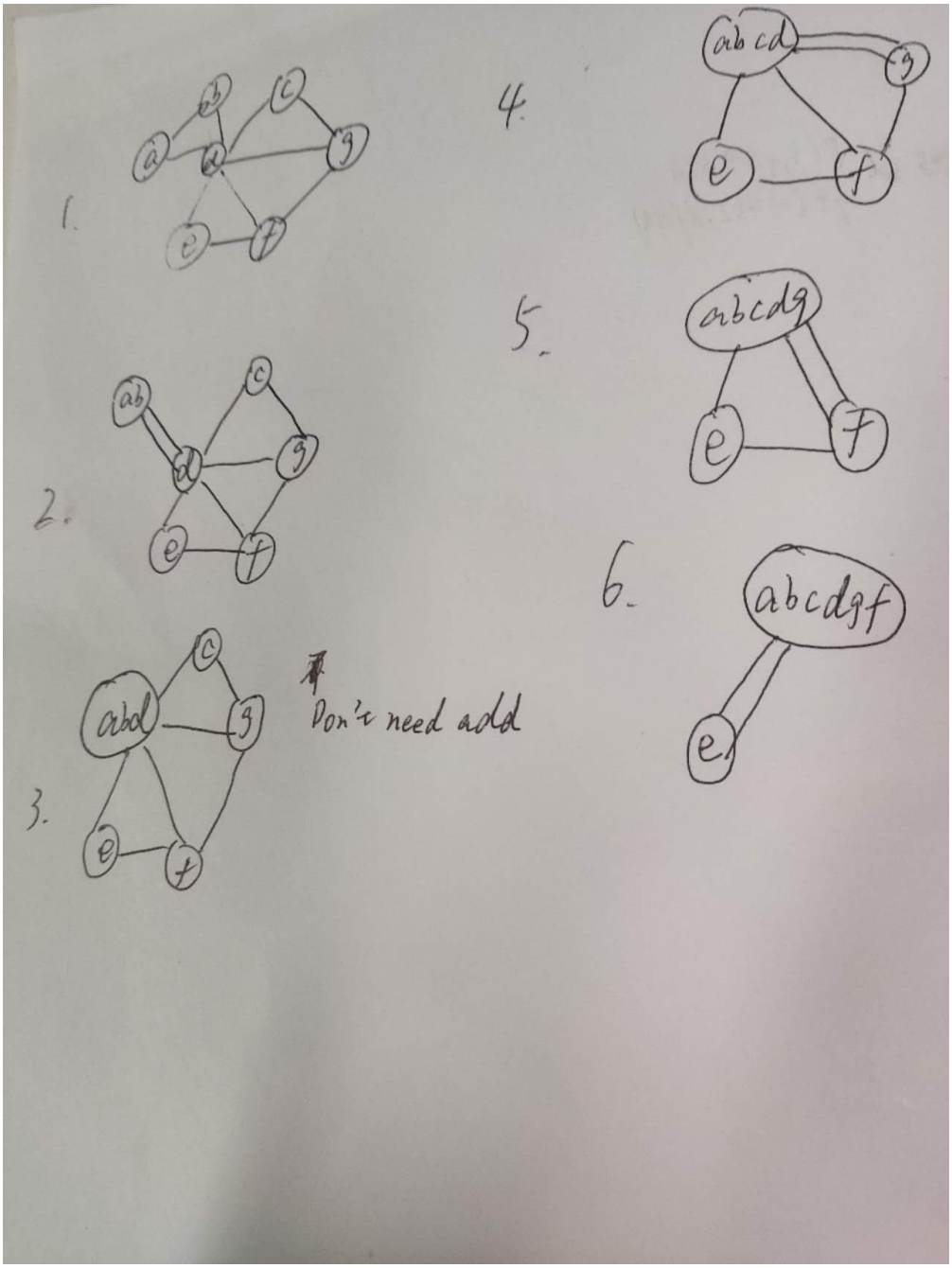


Figure 2: problem 6

9 Problem 9

a) $\forall\alpha, \exists\varphi$, such that $\varphi(\alpha) \in L_2$ and φ a linear-time complexity function. Since, the time complexity of L_2 to solve $\varphi(\alpha)$ is $\Theta\left(n^2\right)$, so the time complexity for L_1 to solve α is $\Theta\left(n^2\right) + O(n) = \Theta\left(n^2\right)$

b) $\forall\beta, \exists\varphi$, such that $\varphi(\beta) \in L_3$ and φ a linear-time complexity function which also needs $O(n)$ plus the time complexity of L_3. Therefore, for $\gamma \in L_3$, it can be solved by mapping back to L_2 with $O(n) + \Theta\left(n^2\right)$, while time complexity for others are not known. So, the time complexity of L_3 is at least n^2. So, the time complexity of L_3 is $\Omega\left(n^2\right)$.

10 Problem 10

Part of the answer is from https://www.geeksforgeeks.org/proof-that-clique-decision-problem-is-np-complete/ .

10. In order to prove the problem is an NP-complete problem, we need to prove it is an NP and an NP-hard problem.

① It is an NP problem.

The certificate is a subset of vertices in G.

The verification is to check whether each pair of vertices in the subset are adjacent by checking if they share an edge between each other. It can be done in $O(|E|+|V|)$

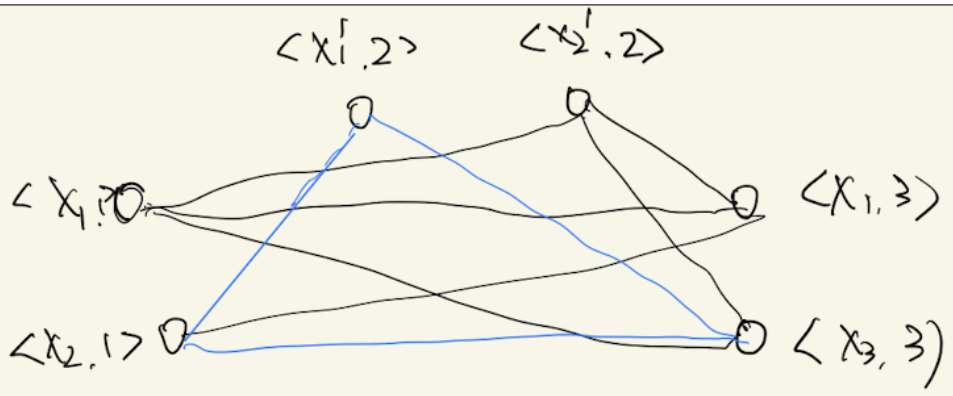
② It is an NP-hard problem.

Proof Boolen Satisfability problem reduce to clique decision Problem.

Let the boolen expression be $F = \underbrace{(x_1 \vee x_2)}_{C_1} \wedge \underbrace{(x_1' \vee x_2')}_{C_2} \underbrace{\wedge (x_1 \vee x_2)}_{C_3}$

We connect these vertices such that

1. No two vertices belongs to the same clause are connect.
2. No variable is connected to its complentere.



Thus the Graph G(V, E) is constructed such that

$V = \{ \langle a, i \rangle \mid a$ belongs to $C_i \}$ and $E = \{ \{ \langle a, i \rangle, \langle b, j \rangle \} \mid$

i is not equal to j; b is not equal to $a' \}$

Consider the subgraph of G with the vertices $\langle x_2, 1 \rangle$, $\langle x_1', 2 \rangle$; $\langle x_3, 3 \rangle$. It forms a clique of size 3.

Corresponding to this, $f(0, 0, 1)$ is true.

So if we have k clauses in our Satisfiability expression, we can get a max clique of size k.

So satisfiability problem is reduced to clique decision problem.

11 Problem 11

For a set cover problem U,F, construct a cover using greedy algorithm: U = 1,2,3,4,5,6,7, F = S1,S2,S3,S4,S5, S1 = 1,2,7, S2 = 2,3,4, S3 = 3,6, S4 = 4,5, S5 = 3,4.

Answer:

All cost is same, we can set as 1.

First iteration:

I =

The per new element cost for S1 = Cost(S1)/|S1 – I| = 1/3

The per new element cost for S2 = Cost(S2)/|S2 – I| = 1/3

The per new element cost for S3 = Cost(S3)/|S3 – I| = 1/2

The per new element cost for S4 = Cost(S4)/|S4 – I| = 1/2

The per new element cost for S5 = Cost(S5)/|S5 – I| = 1/2

Since S1 has minimum value S1 is added, I becomes 1,2,7.

Second iteration:

The per new element cost for S2 = Cost(S2)/|S2 – I| = 1/2

The per new element cost for S3 = Cost(S3)/|S3 – I| = 1/2

The per new element cost for S4 = Cost(S4)/|S4 – I| = 1/2

The per new element cost for S5 = Cost(S5)/|S5 – I| = 1/2

Since S2 has minimum value S2 is added, I becomes 1,2,3,4,7.

Third iteration:

The per new element cost for S3 = Cost(S3)/|S3 – I| = 1/1

The per new element cost for S4 = Cost(S4)/|S4 – I| = 1/1

The per new element cost for S5 = Cost(S5)/|S5 – I| = 1/0

Since S5 has minimum value S3 is added, I becomes 1,2,3,4,6,7.

Fourth iteration:

The per new element cost for S4 = Cost(S4)/|S4 – I| = 1/1

The per new element cost for S5 = Cost(S5)/|S5 – I| = 1/0

Since S5 has a minimum value S4 is added, and I become 1,2,3,4,5,6,7.

Thus, we choose S1,S2,S3, S4.

12 Problem 12

Find the cardinality of a maximum independent set of the following tree using a dynamic programming algorithm.

We need choose from the leaves.

I(k) = 1

I(l) = 1

I(m) = 1

I(h) = max I(k) +I(l), 1 + 0 = 2

I(i) = max I(m) , 1 + 0 = 1

I(j) = 1

I(e) = max I(h) + I(i), 1 + I(k) + I(l) +I(m) = 4

I(f) = 1

I(g) = max 1+ 0, I(j) = 1

I(b) = max I(e) + I(f), 1 + I(h) + I(i) = 5

I(c) = 1

I(d) = max I(g) , 1 + I(j) = 2

I(a) = max I(b) + I(c) +I(d), 1 + I(e) + I(f) + I(g) = 8

So the answer is 8