CSC 226 SUMMER 2023
ALGORITHMS AND DATA STRUCTURES II
ASSIGNMENT 2
UNIVERSITY OF VICTORIA

1. A coin has 1/4 chance of having heads, 3/4 chance of having tails.
   a) What is the expected number of coin tosses needed to get a heads? Why?
   b) Suppose that such a coin is tossed n times. What is the probability that the number of heads equals the number of tails? Write the expression using $\binom{n}{x}$ notation.

2. Draw the hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using hash function $h(k) = (2k + 5) \bmod 11$, assuming collisions are handled by each of the following:
   a) Separate chaining.
   b) Linear probing.
   c) Quadratic probing up to the point where the method fails because no empty slot is found.
   d) Double hashing using the secondary hash function $h'(k) = 7 - (k \bmod 7)$.

3. Prove that any connected, undirected graph has a vertex whose removal, along with its incident edges, will not disconnect the graph by designing a DFS based algorithm to find such a vertex.

4. Let (x, y, w) denote the edge $\{x, y\}$ with weight w. The graph for the parts (a)and (b) is given by nodes $V = \{A, B, C, D, E\}$ and weighted edges
   $(A, B, 7), (A, C, 5), (A, D, 1), (B, C, 4), (B, D, 7), (B, E, 1), (C, E, 3), (D, E, 2)$.

   Show how to construct a minimum spanning tree using Prim's algorithm. List the edges and nodes in order of when the edge is added to the tree. The first node in T is A. Give the initial values of $D(v)$ for each node v. Each time a node is added to T, give the D values which have changed as a result.

5. (a) Show that you can rescale the edge weights of a graph G by adding a positive constant to all of them without affecting the MST.

   (b) Show that Prim's algorithm still work correctly if the graph G contains edges with negative edge weights.

Koki Itagaki V00034?2

1. A coin has 1/4 chance of having heads, 3/4 chance of having tails.
   a) What is the expected number of coin tosses needed to get a heads? Why?
   b) Suppose that such a coin is tossed n times. What is the probability that the number of heads equals the number of tails? Write the expression using $\binom{n}{x}$ notation.

(a) From the formula to find the expected value of a discrete random variable X is

$$E[X] = \sum_x x \, P_r(X=x)$$

I can make the pmf by $P_r(X=x) = \left(\frac{3}{4}\right)^{x-1} \frac{1}{4}$

So, the expected value of X is:

$$E[X] = \Sigma(x \, P_r(X=x))$$
$$= \Sigma\left(k\left(\tfrac{3}{4}\right)^{k-1} \cdot \tfrac{1}{4}\right)$$

→ this is the geometric distribution $[P(X=x) = (1-p)^{x-1} p]$

The expected value for this is $E(X) = \frac{1}{p} = \frac{1}{\frac{1}{4}} = 4$

∴ the expected number of coin tosses needed to get a heads is 4

(b) Let

The probability of having heads $= p(h) = \frac{1}{4}$

The probability of having tails $= p(t) = \frac{3}{4}$

For this question, I use binomial distribution.

$$p(X=x) = \binom{n}{x} P(h)^x \cdot P(t)^{n-x}$$
$$= \binom{n}{x}\left(\tfrac{1}{4}\right)^x \left(\tfrac{3}{4}\right)^{n-x}$$

... Now I'd like to get the probability when x is $\frac{n}{2}$, so the number of heads and tails are the same.

$$= \underline{\binom{n}{\frac{n}{2}}\left(\tfrac{1}{4}\right)^{\frac{n}{2}}\left(\tfrac{3}{4}\right)^{\frac{n}{2}}}$$

2. Draw the hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using hash function $h(k) = (2k + 5) \bmod 11$, assuming collisions are handled by each of the following:
a) Separate chaining.
b) Linear probing.
c) Quadratic probing up to the point where the method fails because no empty slot is found.
d) Double hashing using the secondary hash function $h'(k) = 7 - (k \bmod 7)$.

$k=12$
$h(12)=(24+5) \bmod 11$
$= 7$

$k=44$
$h(44)=(88+5) \bmod 11$
$= 5$

$k=13$
$h(13)=(26+5) \bmod 11$
$= 9$

$k=88$
$h(88)=(176+5) \bmod 11$
$= 181 \bmod 11$
$= 5$

$k=23$
$h(23)=(46+5) \bmod 11$
$= 7$

$k=94$
$h(94)=(188+5) \bmod 11$
$193 \bmod 11$
$= 6$

$k=11$
$h(11)=(22+5) \bmod 11$
$= 5$

$k=39$
$h(39)=(78+5) \bmod 11$
$= 6$

$k=20$
$h(20)=(40+5) \bmod 11$
$= 1$

$k=16$
$h(16)=(32+5) \bmod 11$
$= 37 \bmod 11$
$= 4$

$k=5$
$h(5)=(10+5) \bmod 11$
$= 4$

a)

index	value
0	
1	→ 20
2	
3	
4	→ 16 → 5
5	→ 44 → 88 → 11
6	→ 94 → 39
7	→ 12 → 23
8	
9	→ 13
10	

b)

index	value
0	11
1	39
2	20
3	5
4	16
5	44
6	88
7	12
8	23
9	13
10	94

By using linear probing, I use this equation

$h(k, i) = (h(k) + i) \bmod 11$ that is by this linear probing

$k = 88$

$h(88, 1) = (5 + 1) \bmod 11$
$= 6$

$k = 11$

$h(11, 6) = (5 + 6) \bmod 11$
$= 0$

$k = 23$

$h(23, 1) = (7 + 1) \bmod 11$
$= 8$

$k = 39$

$h(39, 6) = (6 + 6) \bmod 11$

$k = 5$

$h(5, 9) = (4 + 10) \bmod 11$
$= 3$

$k = 94$

$h(94, 4) = (6 + 4) \bmod 11$
$= 10$

$k = 20$

$h(20, 1) = (1 + 1) \bmod 11$
$= 2$

c)

index	value
0	
1	20
2	16
3	11
4	39
5	44
6	88
7	12
8	23
9	13
10	94

By using quadratic probing,

$$h(k, i) = (h(k) + i^2) \bmod 11$$

$k = 88$

$h(88, 1) = (5 + 1^2) \bmod 11$
$= 6$

$k = 11$

$h(11, 3) = (5 + 3^2) \bmod 11$
$= 3$

$k = 23$

$h(23, 1) = (7 + 1^2) \bmod 11$
$= 8$

$k = 39$

$h(39, 3) = (6 + 3^2) \bmod 11$
$= 4$

$k = 94$

$h(94, 2) = (6 + 2^2) \bmod 11$
$= 10$

$k = 16$

$h(16, 3) = (4 + 3^2) \bmod 11$
$= 2$

For $k = 5$

we need to find

$(4 + i^2) = 11$

since we'd like to get

$(4 + i^2) \bmod 11 = 0$

$i^2 = 11 - 4$

$i^2 = 7$

$i = \sqrt{7}$

this doesn't work

(d)

	index	value
✓	0	11
✓	1	23
✓	2	20
✓	3	16
✓	4	39
✓	5	44
✓	6	94
✓	7	12
✓	8	88
✓	9	13
	10	5

By using double hashing, with

$h'(k) = 7 - (k \bmod 7)$

$h(k,i) = (h(k) + i h'(k)) \bmod 11$

$k = 88$

$h(88,1) = (5 + (7 - (88 \bmod 7))) \bmod 11$

$= (5 + (7-4)) \bmod 11$

$= 8$

$k = 20$

$h(20,1) = (1 + (7 - (20 \bmod 7))) \bmod 11$

$= (1 + (7-6)) \bmod 11$

$= 2 \bmod 11 = 2$

$k = 23$

$h(23,1) = (7 + (7 - (23 \bmod 7))) \bmod 11$

$= (7 + (7-2)) \bmod 11$

$= 12 \bmod 11$

$= 1$

$k = 16$

$h(16,2) = (4 + 2(7 - (16 \bmod 7))) \bmod 11$

$= (4 + 2(7-2)) \bmod 11$

$= 14 \bmod 11 = 3$

$k = 11$

$h(11,2) = (5 + 2(7 - (11 \bmod 7))) \bmod 11$

$= (5 + 2(7-4)) \bmod 11$

$= 11 \bmod 11 = 0$

$k = 5$

$h(5,3) = (4 + (7 - (5 \bmod 7))) \bmod 11$

$= (4 + 3(7-5)) \bmod 11$

$= 10 \bmod 11$

$= 10$

$k = 39$

$h(39,3) = (6 + 3(7 - (39 \bmod 7))) \bmod 11$

$= (6 + 3(7-4)) \bmod 11$

$= 15 \bmod 11$

$= 4$

3. Prove that any connected, undirected graph has a vertex whose removal, along with its incident edges, will not disconnect the graph by designing a DFS based algorithm to find such a vertex.

Proof:

I'll prove by contradiction.

Assume that there exists a connected undirected graph where the removal of any vertex and its incident edges will disconnect the graph.

- Let G be a connected, undirected graph with n vertices and because G is connected, there exists a path between any two vertices.

- Now, I use a DFS traversal which starts from an arbitrary vertex v in G. During doing DFS traversal, All vertices which are reachable from v will be visited. I'll call the set of vertices visited as "visited" and remaining vertices which are not visited during the traversal as "remaining".

- Since all vertices in "remaining" are not visited, so If I remove any vertex from "remaining", It will not cause the disconnectivity between vertices in "visited". This means, removing any vertex from "remaining" will not affect "visited".

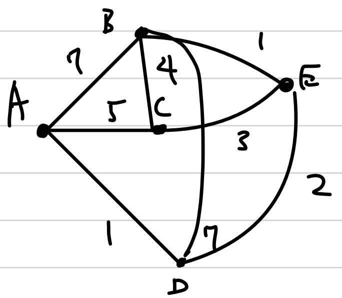
- This contradicts my assumption there exists a connected undirected graph where the removal of any vertex and its incident edges will disconnect the graph. Therefore this assumption isn't correct.

This means it is clear that there exists a vertex whose removal will not disconnect the graph ■

4. Let (x, y, w) denote the edge $\{x, y\}$ with weight w. The graph for the parts (a)and (b) is given by nodes $V = \{A, B, C, D, E\}$ and weighted edges $(A, B, 7), (A, C, 5), (A, D, 1), (B, C, 4), (B, D, 7), (B, E, 1), (C, E, 3), (D, E, 2)$.

Show how to construct a minimum spanning tree using Prim's algorithm. List the edges and nodes in order of when the edge is added to the tree. The first node in T is A. Give the initial values of $D(v)$ for each node v. Each time a node is added to T, give the D values which have changed as a result.

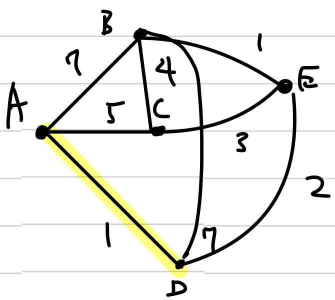
Briefly, I'll explain how prim's theorem works. First choose the arbitrary node and add node to MST. Secound, Consider all the edges connected to the node chosen and find the edge with the minimum weight and add to MST. Thirdly, update the weights based on the nodes added to MST and find a new node with minimum weights and add to MST. Repeat these 3 steps, and get the minimum weight edge until all nodes are included in MST.



① starts with node A as this initial step.

D

A	B	C	D	E
0	7	5	1	+∞

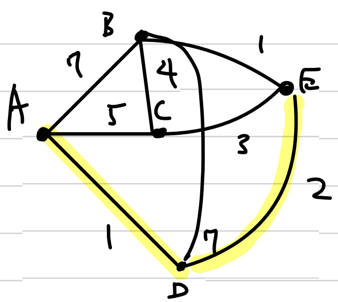


② The edge with the minimum weight is the edge between A and D which is 1. Add the edge to the MST.

D

A	B	C	D	E
0	7	5	1	+∞

The spanning tree (T) is T:{(A, D)}

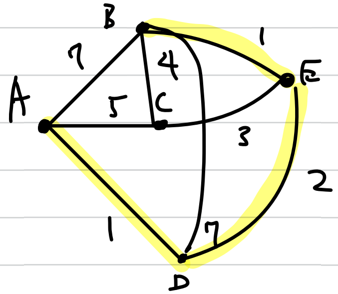


③ The edge with the minimum weight is the edge between D and E which is 2. Add the edge to the MST.

D

A	B	C	D	E
0	7	5	1	+∞
0	7	5	1	2

The spanning tree (T) is T:{(A, D), (D, E)}

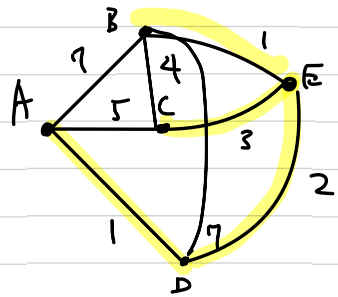


④ The edge with the minimum weight is the edge between B and E which is 1. Add the edge to the MST.

D

A	B	C	D	E
0	7	5	1	+∞
0	7	5	1	2
0	1	3	1	2

The spanning tree (l) is T: (A, D), (D, E), (B, E)



⑤ The edge with the minimum weight is the edge between C and E which is 3. Add the edge to the MST.

D

A	B	C	D	E
0	7	5	1	+∞
0	7	5	1	2
0	1	3	1	2
0	1	3	1	2

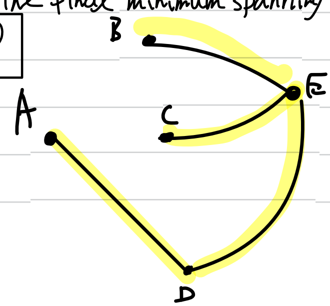
The spanning tree (l) is T: (A, D), (D, E), (B, E) (C, E)

Now all nodes are included and the final minimum spanning tree (T) consists of

the list of edges (A, D), (D, E), (B, E) (C, E)

and the list of nodes A, D, E, B, C

↳ In order when the edge is added.



5. (a) Show that you can rescale the edge weights of a graph G by adding a positive constant to all of them without affecting the MST.

(b) Show that Prim's algorithm still work correctly if the graph G contains edges with negative edge weights.

(a) Assume that the graph G has edges with edge weights $w_1, w_2, w_3 \dots w_n$.
If I rescale all edges weights by adding a positive constant C, I can say that new edge weights would be $w_1+c, w_2+c, w_3+c \dots w_n+c$

First of all, Adding a positive constant to all weights will not affect the relative order of the weights.
For example, There are 2 edges: A and B, and A is less than B. ($A < B$)
If I add a constant C to both edges, the weights of A is still smaller than B's weight Since we added the same constant C to all edges ($c+A < c+B$)

Moreover, The MST is basically determined by the relative order of the edge weights.
The MST usually find the edges with minimum weights not specific value.
As long as the relative order remains The MST will not be changed.

Therefore It is clear that rescaling the edge weights by adding a positive constant to all of edges will not affect the MST as long as the relative order is maintained.
After rescaling, the graph has the same set of edges and form the MST with adjusted weights.

(b) Prim's algorithm work with negative edge weights since the correctness of the algorithm doesn't depend on using positive weights edges.
Let's assume we have subset of MST called T.
T is always maintained to form a single tree. The prim algorithm continuously adds the lightest edge that connects the tree to a vertex outside the tree. It is important that negative weights does not have any impact on the process which finds the lightest edge.
So, the algorithm's correctness isn't affected by negative weights since the focus is on finding the lightest edge, regardless of its sign
Therefore, negative weights don't pose a problem in Prim's algorithm.