CSCI 3104: Algorithms
Spring 2022
Recitation #7 - Midterm Review

Midterm TA reviews. Please take take a few minutes to fill out TA FCQs! They're very helpful for us TAs. You can find the link in your email under the subject line "Computer Science Midterm TA FCQ's."

Problem 1

Prove the following via induction (strong or weak).

a. $\sum_{k=1}^{n} q^{k-1} = \frac{q^n - 1}{q-1}$, for all $n \geq 1$, $q \neq 1$.

b. $\sum_{k=1}^{n}(2k-1) = n^2$ for all $n \geq 1$.

c. (Bonus.) Any integer $n \geq 2$ is a product of prime numbers.

Problem 2

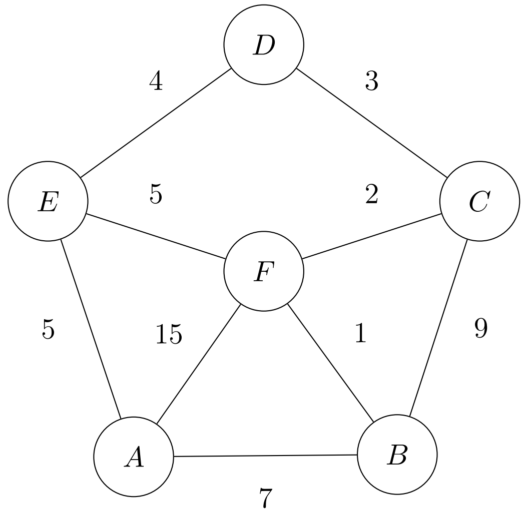
Construct a Huffman Code for the phrase "she sells sea shells". The frequencies for each letter are as follows (we ignore spaces):

letter	a	e	h	l	s
number of ocurrences	1	4	2	4	6
frequency	0.06	0.24	0.12	0.23	0.35

(*Note:* "e" and "l" actually have the same frequency, but we round one up and one down to give each letter a distinct frequency in this example.)

Problem 3

Run Dijkstra's algorithm on the following graph to find a SSSP tree, starting from vertex A.



Problem 4

You've decided to go on a backpacking trip in the beautiful Rocky Mountains. Your itinerary is pretty ambitious, and you can't waste time stopping to dig through your bag to get things out. Luckily, you can use your algorithms expertise to pack your bag as efficiently as possible, to minimize the amount of time you spend rooting around for something at the bottom.

To be concrete, you have n items to pack. Item i has length $\ell_i \geq 0$ and probability p_i of you wanting to access it. Assuming you can only pack items on top of each other, and that the time to access the i^{th} item down your pack is directly proportional to the combined lengths of the top i items, determine which order to place the items such that the *expected* access time T is minimized. In other words, minimize

$$T = \sum_{i=1}^{n} p_i L_i,$$

where L_i is the combined length of items above and including i in the pack.

a. Devise an algorithm to order your gear in a way that minimizes the total expected access time.

b. Prove your algorithm is correct.

Problem 1

Prove the following via induction (strong or weak).

a. $\sum_{k=1}^{n} q^{k-1} = \frac{q^n-1}{q-1}$, for all $n \geq 1$, $q \neq 1$.

BC: when $n=1$, $\sum_{k=1}^{1} q^{k-1} = q^0 = 1$

$\frac{q^n-1}{q-1} = \frac{q-1}{q-1} = 1$

IH: for some $m \geq 1$, $\sum_{k=1}^{m} q^{k-1} = \frac{q^m-1}{q-1}$

IS: Show $\sum_{k=1}^{m+1} q^{k-1} = \frac{q^{m+1}-1}{q-1}$ as follows.

$\sum_{k=1}^{m+1} q^{k-1} = q^{m+1-1} + \sum_{k=1}^{m} q^{k-1} \overset{\text{using IH.}}{=} q^m + \frac{q^m-1}{q-1}$

$= \frac{q^m(q-1)+q^m-1}{q-1}$

$= \frac{q^{m+1}}{q-1}$

b. $\sum_{k=1}^{n}(2k-1) = n^2$ for all $n \geq 1$.

BC: $n=1$. $\sum_{k=1}^{1}(2k-1) = 2-1 = 1$, $n^2 = 1^2 = 1$ $\Big\}$ $\sum_{k=1}^{n}(2k-1) = n^2$ when $n=1$ #

IH: for some $m \geq 1$, $\sum_{k=1}^{m}(2k-1) = m^2$

IS: Show $\sum_{k=1}^{m+1}(2k-1) = (m+1)^2$ as follows:

$\sum_{k=1}^{m+1}(2k-1) = 2(m+1)-1 + \sum_{k=1}^{m}(2k-1)$

using IH $= 2m+1 + m^2$

$= (m+1)^2$

we show that property holds for the case $m+1$. by IH.

#

c. (Bonus.) Any integer $n \geq 2$ is a product of prime numbers.

BC. $n=2$, ...

IH. Assume the statement holds for all $2 \leq k \leq n$, for some n.

IS. We show that statement holds for $n+1$, as well.

If $n+1$ is prime, ...

$n+1$ isn't prime, then $n+1 = a \cdot b$, where

$2 \leq a \leq n$ and $2 \leq b \leq n$

Thus, ...

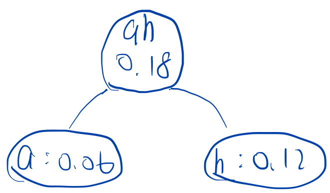
Problem 2

Construct a Huffman Code for the phrase "she sells sea shells". The frequencies for each letter are as follows (we ignore spaces):

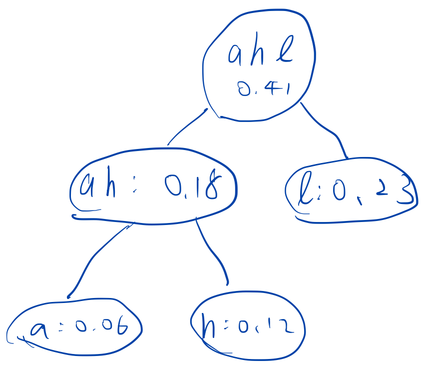
letter	a	e	h	l	s
number of ocurrences	1	4	2	4	6
frequency	0.06	0.24	0.12	0.23	0.35

(*Note:* "e" and "l" actually have the same frequency, but we round one up and one down to give each letter a distinct frequency in this example.)

a	h	l	e	s
0.06	0.12	0.23	0.24	0.35



ah	l	e	s
0.18	0.23	0.24	0.35



e	s	ahl
0.24	0.35	0.41

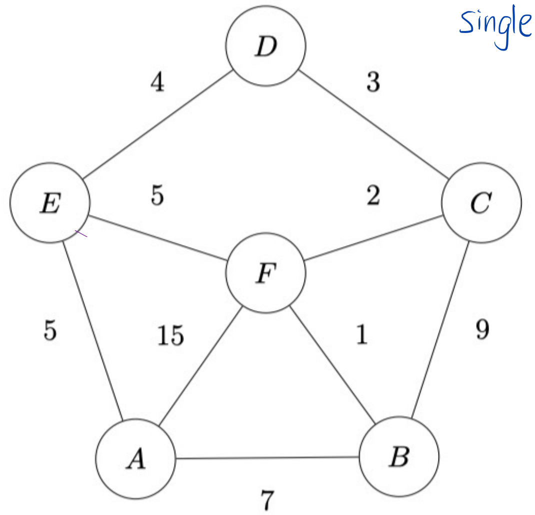
Note

```
HUFFMAN(C)
n = |C|
Q = C
for i = 1 to n - 1
    allocate a new node z
    z.left = x = EXTRACT-MIN(Q)
    z.right = y = EXTRACT-MIN(Q)
    z.freq = x.freq + y.freq
    INSERT(Q, z)
return EXTRACT-MIN(Q)        // return the root of the tree
```

Problem 3

Run Dijkstra's algorithm on the following graph to find a SSSP tree, starting from vertex A.

single source shortest path.



	A	B	C	D	E	F
poll A.	0	∞	∞	∞	∞	∞
E	-	7	∞	∞	5	15
B	-	7	∞	9		10
F			16	9		8
D	-		10	9		
C			10			

$Q = \{(A,0)\}$

$Q = \{(E,5), (B,7), (F,15)\}$

$\{(B,7), (D,9), (F,10)\}$

$\{(F,8), (D,9), (C,16)\}$

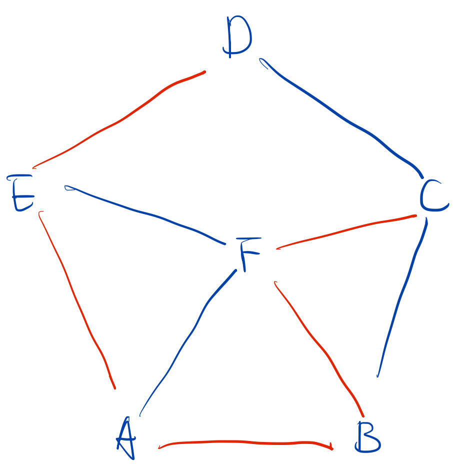
$\{(D,9), (C,10)\}$

$\{(C,10)\}$

$\{\ \}$

Queue is empty now!

Done!



Problem 4

You've decided to go on a backpacking trip in the beautiful Rocky Mountains. Your itinerary is pretty ambitious, and you can't waste time stopping to dig through your bag to get things out. Luckily, you can use your algorithms expertise to pack your bag as efficiently as possible, to minimize the amount of time you spend rooting around for something at the bottom.

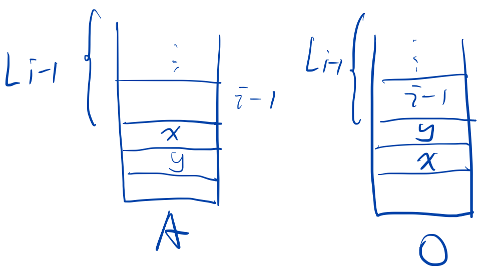
To be concrete, you have n items to pack. Item i has length $\ell_i \geq 0$ and probability p_i of you wanting to access it. Assuming you can only pack items on top of each other, and that the time to access the i^{th} item down your pack is directly proportional to the combined lengths of the top i items, determine which order to place the items such that the *expected* access time T is minimized. In other words, minimize

$$T = \sum_{i=1}^{n} p_i L_i, \quad = \sum_{i=1}^{n} p_i \sum_{i=1}^{n} l_i$$

where L_i is the combined length of items above and including i in the pack.

a. Devise an algorithm to order your gear in a way that minimizes the total expected access time.
We use the greedy by largest p_i /l_i ratio algorithm, with items deeper in the bag having smaller ratio.

b. Prove your algorithm is correct.



From the def

$$\frac{p_x}{l_x} \geq \frac{p_y}{l_y} \Rightarrow p_x \cdot l_y \geq p_y \, l_x$$

$$p_x(L_{i-1} + l_x) + p_y(L_{i-1} + l_x + l_y)$$

$$\leq p_x(L_{i-1} + l_x + l_y) + p_y(L_{i-1} + l_y)$$

Hence, the total cost will be greater or stay equal if exchanging items from the suggested order by the greedy algorithm.