Solver: Chew Jing Wei

Email Address: jchew029@e.ntu.edu.sg

1a

i. True
ii. True
iii. True
iv. True
v. True because if any NP-complete problem can be reduced to P, that means all NP problems are reducible to P as well, and thus P = NP.

1b

i. Best case: searched key is the root. Number of key comparisons = 1.

Worst case: searched key is a leaf. Number of key comparisons = $\lceil n \rceil$

ii. Number of key comparisons at each level = level index in the binary search tree * 2 (starting from root as index 1)

So, average-case time complexity = $\frac{2 * \sum_{i=1}^{k} \quad i}{k} = \frac{k * (k+1)}{k} = k + 1$

(Explanation: there is an equal chance for the key to be found at each level. Thus, the average will be the summation of all possible number of comparisons (from level = 1 to level = k) divided by the number of levels, multiplied by 2 since there are 2 key comparisons per non-key level)

1c

Solution in Python:

```
def recSumSquare(n):
  if n == 1:
    return 1
  else:
    return n**2 + recSumSquare(n - 1)
```

1d

i. T(n) = k * T(n / c) + f(n)
ii. T(n) = 4 * T(n / 2) + n

= 4^2 * T (n / 4) + n + 2n

= 4^3 * T (n / 8) + n + 2n + 4n

= 4^4 & T (n / 16) + n + 2n + 4n + 8n

= ...

= 4^n * T(1) + n * Sum of G.P,

where r = 2, initial term = 1 and number of terms = n

$= n * (2^n - 1)$

2a

i. $\frac{f(n)}{g(n)} = \frac{n}{n^2} = \frac{n}{2^{\frac{n}{4}}} = 1$

Hence, f(n) is in O(g(n)), and f(n) is in $\Omega(g(n))$, and f(n) is in $\theta(g(n))$

ii. $\frac{f(n)}{g(n)} = \frac{a^{n+1}}{(a+1)^n} = \frac{a^n * a}{(a+1)^n} = \left(\frac{a}{a+1}\right)^n * a = 0$

Hence, f(n) is in O(g(n)), but f(n) is not in $\Omega(g(n))$, and f(n) is not in $\theta(g(n))$

2b

i. Worst case scenario is when every key within the sequence initially gets hashed to the same slot on the hash table, and thus form a congruent block on it when inserted.

Total number of comparisons = $0 + 1 + 2 + .. + (n - 1) = \frac{(n-1) * n}{2}$

ii. $\frac{1}{h} * \sum_1^{\frac{n}{4}} i$ because there are h slots, so equal probability of being slotted into each of them. Then, $\sum_1^{\frac{n}{4}} i$ is the total number of possible collisions.

2c

The decision problem is to determine whether there is a subset of objects that can fit into C and has total profit of at least k.

The problem is in NP because given a subset of the object, there is a polynomial time algorithm on a non-deterministic computer to verify if the total profit is at least k.

3a)

1st outer loop iteration (pointing at 18):

- 1 comparison
- 0 swap

2nd outer loop iteration (pointing at 7):

- 2 comparisons
- 1 swap

3rd outer loop iteration (pointing at 34):

- 1 comparison
- 0 swap

4th outer loop iteration (pointing at 7):

- 3 comparisons
- 2 swaps

5th outer loop iteration (pointing at 2):

- 5 comparisons
- 4 swaps

Total number of comparisons = 12

Total number of swaps = 7

3b

Worst case for a 2-way merge function is n-1.

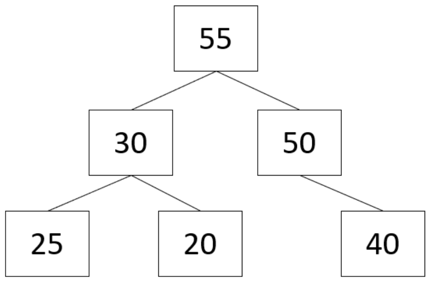
So, worst case for the 3-way merge function (which invokes the 2-way merge function as a subroutine) is (2n/3 – 1) + (n – 1). (2n/3 – 1) is the worst-case number of comparisons required to merge the first two arrays, which have a combined size of 2n/3.

3c

2, 4, 1, 3, 5

Thought process: For the subarray to always be empty, the element chosen as the pivot must always be the smallest. Working towards this objective, think of the possible permutations that may achieve this. In the exam, it took me about 3 tries.

3d

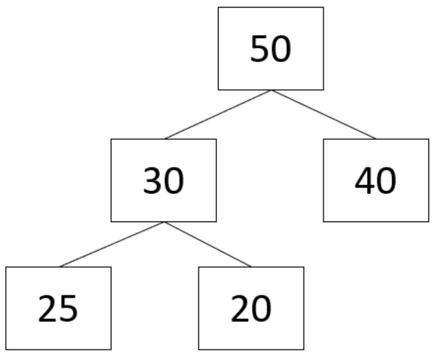


i.

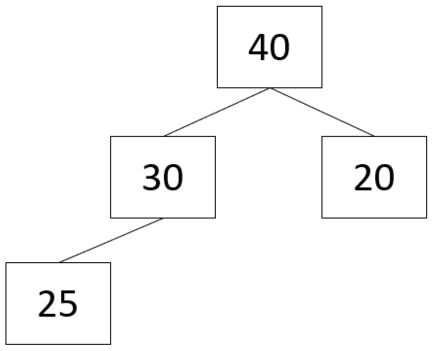
7 comparisons, 4 swaps

ii.

After first deleteMax() call,

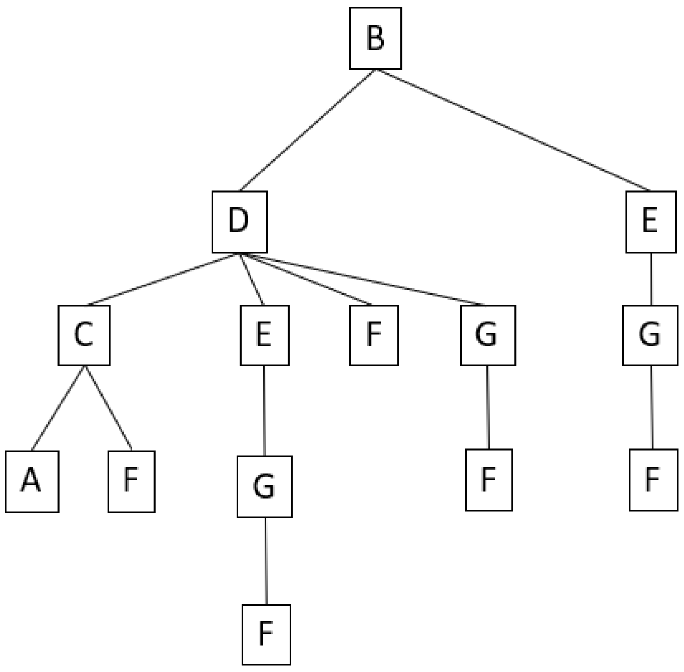


After second deleteMax() call,



4a

i.



ii.

vertex	distance	Predecessor	Path

A	9	C	B -> D -> C -> A
B	-	-	B
C	5	D	B -> D -> C
D	3	B	B -> D
E	5	D	B -> D -> E
F	8	G	B -> D -> G -> F
G	7	D	B -> D -> G

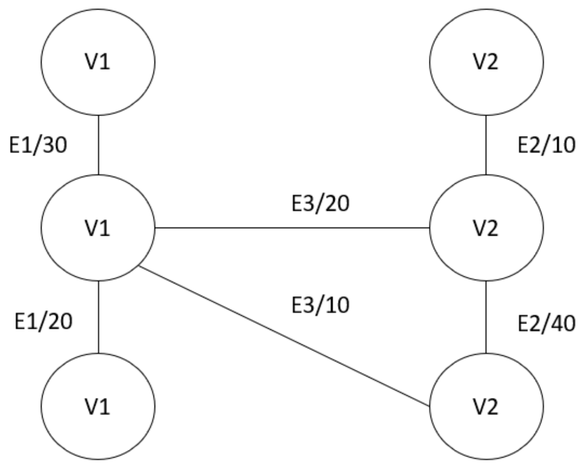
Trick to this is to draw the graph out manually and get your shortest paths by visual inspection.

4b. Yes. Prim's algorithm does not care about the sign of the weights in the input graph – it works by taking the next minimum weight edge and having negative weights will not affect the correctness of the algorithm when picking a minimum value.

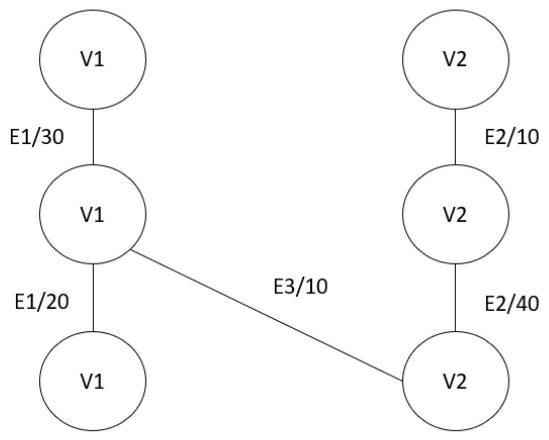
4c. No, it won't.

Counter example:

Consider this graph:



Going by the logic given in the question, the spanning tree formed will be:



Which is not a minimum spanning tree.

All the best for your exams!