Computer Science 477/677 Spring 2019

University of Nevada, Las Vegas Computer Science 477/677 Spring 2019

Answers for Practice for Second Examination March 13, 2019

1. Review Assignment 3.

2. Construct a treap with alphabetic key and numeric min-heap order. You are to insert the items one at a time and show the treap after each rotation. Insert letters in this order: A, B, C, D, E, F. The numeric heap keys (the random numbers) are given in the following table.

A	23
B	12
C	11
D	7
E	4
F	1

3. Find the asymptotic complexity, in terms of n, for each of these fragments, expressing the answers using O, Θ, or Ω, whichever is most appropriate.

   (a)
   ```
   for(i = 0; i < n; i = i+1);
    cout << "Hi!" << endl;
   ```
   $\Theta(n)$

   (b)
   ```
   for(i = 1; i < n; i = 2*i);
    cout << "Hi!" << endl;
   ```
   This reduces to Problem 3a Substitute $j = \log_2 i$, $m = \log_2 n$. Taking the base 2 logarithm of all the variables of 3b, and substituting, we obtain

   ```
   for(log i = log 1; log i < log n; log i = log i + 1)
   for(j = 0; j < m; j = j+1)
   ```

   Which is 3a. The solution is $\Theta(m) = \Theta(\log n)$.

   (c)
   ```
   for(i = 2; i < n; i = i*i);
    cout << "Hi!" << endl;
   ```
   This reduces to Problem 3b Substitute $j = \log_2 i$, $m = \log_2 n$. Taking the base 2 logarithm of all the variables of 3b, and substituting, we obtain

   ```
   for(log i = log 2; log i < log n; log i = log (i*i))
   for(log i = log 2; log i < log n; log i = 2*log i)
   for(j = 1; j < m; j = 2*j)
   ```

   Which is 3b. The solution is $\Theta(\log m) = \Theta(\log \log n)$.

   (d) The following code models the first phase of heapsort.

   ```
   for(int i = n; i > 0; i--)
    for(int j = i; 2*j <= n; j = 2*j)
     cout << "swap" << endl;
   ```

$\Theta(n)$

(e) The following code models the second phase of heapsort.

```
for(int i = n; i > 0; i--}
 {
  cout << "swap" << endl;
  for(int j = 1; 2*j <= i; j = 2*j)
   cout << "swap" << endl;
 }
```

$\Theta(n \log n)$

(f) The following code models insertion of n items into an AVL tree.

```
for(int i = 1; i < n; i++)
 for(int j = n; j > 0; j = j/2)
   cout << "check AVL property and possibly rotate" << endl;
```

$\Theta(n \log n)$

4. Solve each of the following recurrences, expressing the answers using O, Θ, or Ω, whichever is most appropriate.

   (a) $F(n) = F(n/2) + 1$

   $\Theta(\log n)$ by the master theorem

   (b) $F(n) = F(n-1) + O(\log n)$

   $\Theta(n \log n)$ by the anti-derivative method

   (c) For this problem, as well as the next one, see

   `https://en.wikipedia.org/wiki/Akra%E2%80%93Bazzi_method#Example`

   $F(n) = F\left(\frac{n}{2}\right) + 2F\left(\frac{n}{4}\right) + n$

   $C = 1$ and $\left(\frac{1}{2}\right)^C + 2\left(\frac{1}{4}\right)^C = 1$ and therefore the complexity is $\Theta(n \log n)$.

   (d) $F(n) = F\left(\frac{3n}{5}\right) + F\left(\frac{4n}{5}\right) + n^2$

   $C = 2$ and $\left(\frac{3}{5}\right)^C + \left(\frac{4}{5}\right)^C = 1$ and therefore the time complexity is $\Theta(n^2 \log n)$

   (e) $F(n) = F(n-2) + n$

   $\Theta(n^2)$ by the anti-derivative method

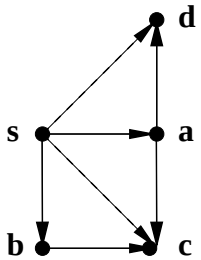
5. Use Huffman's algorithm to construct an optimal prefix code for the alphabet $\{A, B, C, D, E, F\}$ where the frequencies of the symbols are given by the following table.

A	7
B	3
C	4
D	8
E	12
F	3

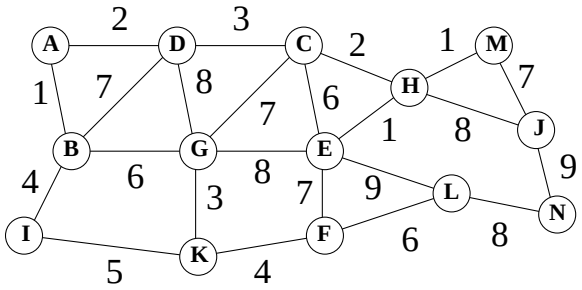
6. G is an acyclic directed graph with vertices $\{s, a, b, c, d\}$. G has exactly five topological orderings, namely

   s, a, b, c, d
   s, a, b, d, c
   s, a, d, b, c
   s, b, a, c, d
   s, b, a, d, c

   There is more than one answer. Some arcs must exist, some arcs must not exist, and some arcs are optional. Optional arcs are shown as dashed arrows.



7. Find a minimum spanning tree of the weighted graph shown below.



Use union/find, with path compression.

8. Insert the letters A, B, C, D, E, F into an AVL tree in that order. Show the rotations (if any) after each insertion.

9. Write pseudo-code for binary search.

10. Walk through heapsort for the following array: **A Q R B X S M L N T**

Items which are done are shown in boldface.

1	2	3	4	5	6	7	8	9	10
A	Q	R	B	X	S	M	L	N	T
A	Q	R	N	X	S	M	L	B	T
A	Q	S	N	X	R	M	L	B	T
A	X	S	N	Q	R	M	L	B	T
A	X	S	N	T	R	M	L	B	Q
X	A	S	N	T	R	M	L	B	Q
X	T	S	N	A	R	M	L	B	Q
X	T	S	N	Q	R	M	L	B	A
A	T	S	N	Q	R	M	L	B	**X**
T	A	S	N	Q	R	M	L	B	**X**
T	Q	S	N	A	R	M	L	B	**X**
B	Q	S	N	A	R	M	L	**T**	**X**
S	Q	B	N	A	R	M	L	**T**	**X**
S	Q	R	N	A	B	M	L	**T**	**X**
L	Q	R	N	A	B	M	**S**	**T**	**X**
R	Q	L	N	A	B	M	**S**	**T**	**X**
R	Q	M	N	A	B	L	**S**	**T**	**X**
L	Q	M	N	A	B	**R**	**S**	**T**	**X**
Q	L	M	N	A	B	**R**	**S**	**T**	**X**
Q	N	M	L	A	B	**R**	**S**	**T**	**X**
B	N	M	L	A	**Q**	**R**	**S**	**T**	**X**
N	L	M	B	A	**Q**	**R**	**S**	**T**	**X**
A	L	M	B	**N**	**Q**	**R**	**S**	**T**	**X**
M	L	A	B	**N**	**Q**	**R**	**S**	**T**	**X**
B	L	A	**M**	**N**	**Q**	**R**	**S**	**T**	**X**
L	B	A	**M**	**N**	**Q**	**R**	**S**	**T**	**X**
A	B	**L**	**M**	**N**	**Q**	**R**	**S**	**T**	**X**
B	A	**L**	**M**	**N**	**Q**	**R**	**S**	**T**	**X**
A	**B**	**L**	**M**	**N**	**Q**	**R**	**S**	**T**	**X**