University of Nevada, Las Vegas Computer Science 477/677 Fall 2023

Answers to Third Examination November 22, 2023

1. Fill in the blanks.

   (a) Name two greedy algorithms introduced in class this semester. **Huffman's Kruskal's** .

   (b) In closed hashing, collisions are resolved by the use of **probe** sequences.

   (c) In open hashing, each table position must hold a **search** structure.

2. Give the asymptotic time complexity in terms of n, using Θ, O, or Ω, whichever is most appropriate.

   (a) The worst case time complexity of quicksort on a list of length n.

   $O(n^2)$

   (b) The average case time complexity of quicksort on a list of length n, if pivots are chosen at random.

   $\Theta(n \log n)$

   (c) The worst case time complexity of building a treap with n items.

   $O(n^2)$

   (d) The average case time complexity of building a treap with n items.

   $\Theta(n \log n)$

   (e) Solve the recurrence: $H(n) < 4H(2n/5) + H(3n/5) + 2n^2$

   $4(2/5)^2 + (3/5)^2 = 1$. Therefore $H(n) = \Theta(n \log n)$.

   (f) Solve the recurrence: $G(n) = 4(G(n/2) + 5n^2$

   $4(1/2)^2 = 1$, therefore $G(n) = \Theta(n \log n)$.

   (g) $F(n) = F(n - \log n) + \log^2 n$

   $\frac{F(n) - F(n - \log n)}{\log} n = \frac{\log^2 n}{\log n}$
   $F'(n) = \Theta(\log n)$
   $F(n) = \Theta(n \log n)$

3. For each of these recursive subprograms, write a recurrence for the time complexity, then solve that recurrence.

   (a)
```
void george(int n)
 {
  if(n > 0)
   {
    for(int i = 0; i < n; i++) cout << "hello" << endl;
    george(n/2); george(n/3);
   }
```

```
 }
```

$T(n) = T(n/2) + T(n/3) + n$

$T(n) = \Theta(n)$

(b)
```
void martha(int n)
 {
  if(n > 0)
   for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j++)
     cout << "hello world";
   martha(n/2);
 }
```

$T(n) = T(n/2) + n^2$

$T(n) = \Theta(n^2)$

4. A 3-dimensional $10 \times 20 \times 12$ rectangular array A is stored in main memory in column major order, and its base address is 1024. Each item of A takes two words of main memory, that is, two addressed location. Find the address, in main memory, of A[5][13][7].

   $1024 + 2 * (7 * 20 * 10 + 13 * 10 + 5) = 4094$

5. You are trying to construct a cuckoo hash table of size 8, where each of the 8 names listed below has the two possible hash values indicated in the array. Put the items into the table, if possible. Instead of erasing ejected items, simply cross them out, so that I can tell that you worked it properly.

	h1	h2
Ann	1	0
Bob	4	2
Cal	0	7
Dan	1	6
Eve	1	0
Fay	6	2
Gus	5	3
Hal	3	7

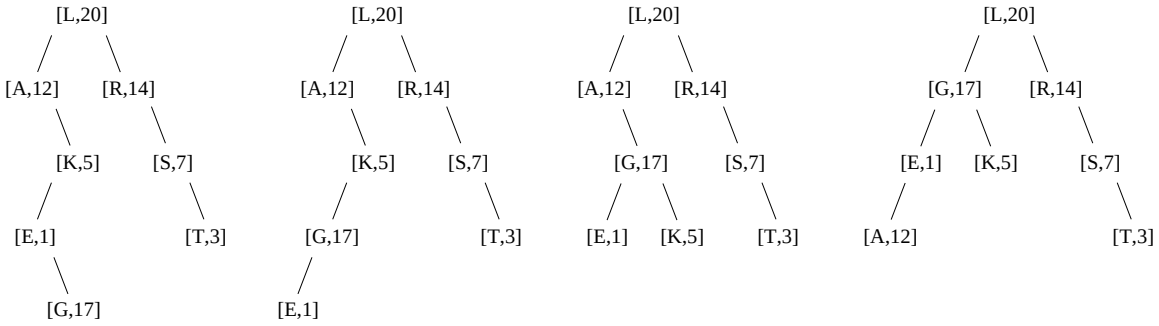
0	~~Cal~~ ~~Ann~~ Eve
1	~~Ann~~ ~~Dan~~ ~~Eve~~ ~~Dan~~ Ann
2	Fay
3	Hal
4	Bob
5	Gus
6	~~Dan~~ ~~Fay~~ Dan
7	Cal

6. Let $\sigma = x_1, x_2, \ldots x_n$ be a sequence of numbers with both positive and negative terms. Write an $O(n)$ time dynamic program which finds maximum sum of any contiguous subsequence of σ. For example, if the sequence is $-1, 4, -3, 2, 7, -5, 3, 4, -8, +6$ then the answer is $4 - 3 + 2 + 7 - 5 + 3 + 4 = 12$.

   $X[i]$ = sum of best subsequence of $x_1, \ldots x_i$
   $Y[i]$ = sum of best subsequence of $x_1, \ldots x_i$ which includes x_i

The program is:
$X[0] = 0$
$Y[1] = x_1$
$X[1] = max\{0, x_1\}$
For i from 2 to n
{
$Y[i] = x_i + max\{0, Y_{i-1}\}$
$X[i] = max\{Y[i], X[i-1]\}$
}

7. The figure below shows a treap, where the data are letters and the nodes of the tree are memos, where the first component is the key, a letter, and the second component is a the priority, a random integer. Insertion of the letter G, where the priority is chosen (at random) to be 17. Show the steps.



8. Explain how to implement a sparse array using a search structure.

Let A be the sparse virtual array. Let S be a search structure which contains ordered pairs of the form (i, x), where $A[i] = x$. To fetch the value of $A[i]$, search S for a pair (i, x). If that pair is found, return x, otherwise return a default value, such as 0. To store a value x for $A[i]$, search S for a pair (i, y). If that pair is found, replace y by x. That pair is not found, insert the pair (i, x) into S.

9. Perform these calculations using complex numbers.

(a) What is the modulus of $e^{\frac{2\pi\mathbf{i}}{7}}$?

The answer is 1, since any imaginary power of e is on the unit circle and hence has modulus 1.

(b) Write $\dfrac{3+4\mathbf{i}}{2+\mathbf{i}}$ in $a + b\,\mathbf{i}$ form.

$$\frac{3+4\mathbf{i}}{2+\mathbf{i}} = \frac{(3+4\mathbf{i})(2-\mathbf{i})}{(2+\mathbf{i})(2-\mathbf{i})} = \frac{10+5\mathbf{i}}{5} = 2+\mathbf{i}.$$

(c) Write $e^{\frac{\pi\mathbf{i}}{8}}$ in $a + b\,\mathbf{i}$ form, without using any trigonometric functions.
(Hint: Use the formula for the square root of a complex number.)

$e^{\frac{\pi\mathbf{i}}{8}}$ is the principle 16th root of unity, which is one of the two square roots of the principle 8th root of unity, $e^{\frac{\pi\mathbf{i}}{4}} = \dfrac{1+\mathbf{i}}{\sqrt{2}}$.

The real and imaginary parts of the principle 8th root of unity are $a = b = \dfrac{1+\mathbf{i}}{\sqrt{2}}$. The square roots are then

$$\pm\left(\sqrt{\frac{\sqrt{a^2+b^2}+a}{2}} + \mathbf{i}\sqrt{\frac{\sqrt{a^2+b^2}-a}{2}}\right)$$

Only one of those is correct, namely

$$\sqrt{\frac{\sqrt{a^2+b^2}+a}{2}} + \mathbf{i}\sqrt{\frac{\sqrt{a^2+b^2}-a}{2}} = \frac{\sqrt{\sqrt{2}+1}+\mathbf{i}\sqrt{\sqrt{2}-1}}{2^{3/4}}$$

10. Consider the following two C++ subprograms.

```
int f(int n)
  {
   if(n > 0)
    return f(n/2)+f(n/3)+f[n/6]+n*n;
   else
    return 0;
  }

void computef(int n)
 {
  f[0] = 0;
  for(int i = 0; i <= n; i++)
   f[i] = f[i/2]+f[i/3]+f[1/6]+n*n;
 }
```

(a) The first of those subprograms is a recursive function. What is the asymptotic value of f(n)?

$\Theta(n^2)$

(b) What is the asymptotic time complexity of the computation of f(n) using the recursive function?

$\Theta(n)$

(c) The second subprogram stores values in an array. What is the asymptotic time complexity of that computation?

$\Theta(n)$

(d) What is the asymtotic time complexity of a computation of f(n) using memoization? (Hint: it's a power of log n.) $\Theta(\log^2 n)$

Walk through Kruskal's algorithm to find the minimum spanning tree of the weighted graph shown below. Show the evolution of the union/find structure. Whenever there is choice between two edges of equal weight, choose the edge which has the alphabetically largest vertex. Whenever there is a union of two trees of equal weight, choose the alphabetically larger root to be the root of the combined tree. Indicate path compression when it occurs.