CS 477/677 Study Guide for Examination November 20, 2024

You will need extra paper to work these problems.

1. Hashing

   (a) What is closed hashing?

   (b) What is open hashing?

   (c) What is open addressing?

   (d) What is a perfect hash function?

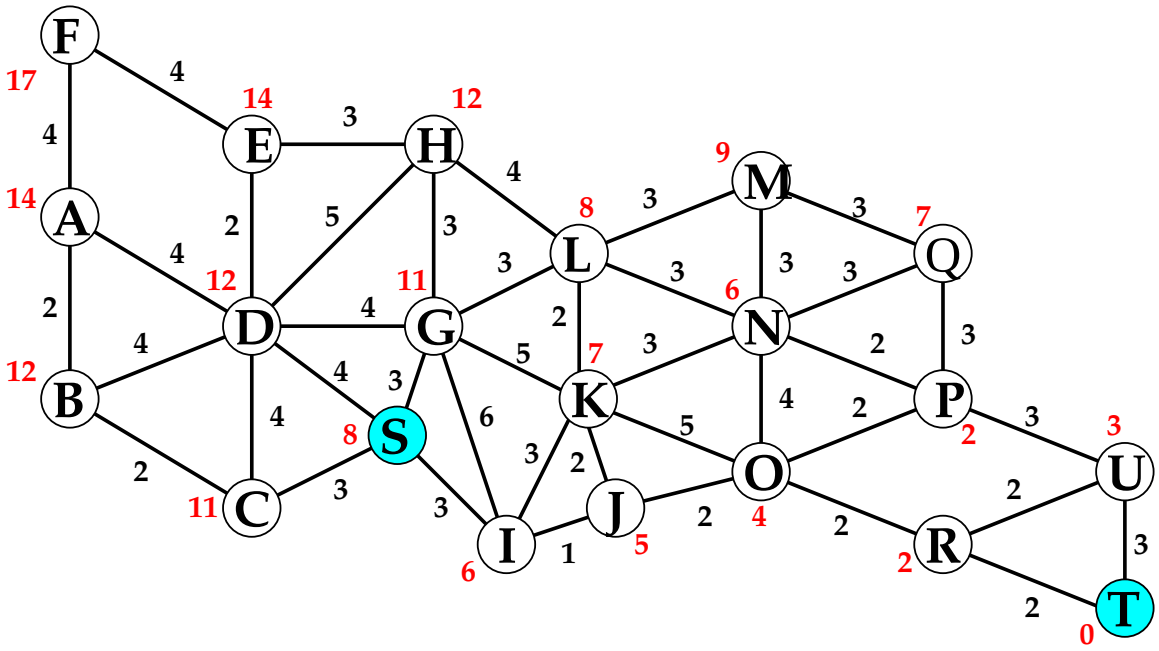
   (e) What are the important properties of a good hash function?

   (f) What is cuckoo hashing?

   (g) Compute the indicated mod 2 matrix product.

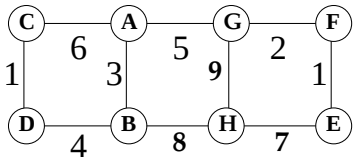
$$\begin{matrix} 10111100 \\ 11010110 \\ 00001011 \\ 00011110 \\ 00111010 \end{matrix} \times \begin{matrix} 111 \\ 101 \\ 001 \\ 010 \\ 100 \\ 100 \\ 011 \\ 101 \end{matrix} =$$

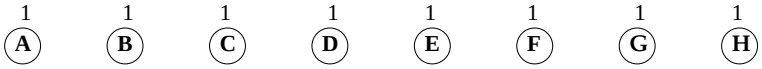
2. Work the A* algorithm for the following weighted diagraph, where the heuristic values are given in red.



3. 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.





4. Fill in the blanks.

   (a) Name two greedy algorithms introduced in class this semester. _____ _____

   (b) In closed hashing, collisions are resolved by the use of _____sequences.

   _____ hashing does not have collisions.

   (c) In closed hashing, if a collision occurs, a _____ _____ can be used to locate an unused position in the hash table.

   (d) In a _____ hash table, each item has two or more possible locations, and must be stored in one of those.

   (e) _____ Which of the following three statements is closest to the truth?
   (1) In SHA256 hashing, collisions are impossible.
   (2) In SHA256 hashing, collisions occur only a few times a year in practice.
   (3) In SHA256 hashing, collisions are so unlikely that industry experts claim they never occur.

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

   _____

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

   _____

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

   _____

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

   _____

   (j) In an open hash table of size m holding n data items, the items at each index of the table are typically shown as linked list. However, that structure is only efficient if each list is of moderate size. In general, we use a _____ at each table index.
   Pick one of these answers:
   heap
   stack
   search structure

(k) A directed graph is defined to be _____ if, given any two vertices x and y, the graph contains a path from x to y.

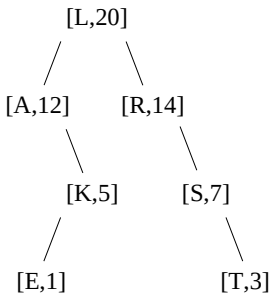
5. A 3-dimensional $10 \times 20 \times 12$ rectangular array A is stored in main memory in column major order, and its base address is 1024. Each item of A takes two words of main memory, that is, two addressed location. Find the address, in main memory, of A[5][13][7].

6. You are trying to construct a cuckoo hash table of size 8 holding 8 names. Each of the names listed below has two possible hash values, as indicated in the array. Put the items into the table in alphabetic order, if possible. Instead of erasing ejected items, simply strike them out.

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

0	
1	
2	
3	
4	
5	
6	
7	

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.

9. 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 the function f computed by the code?

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

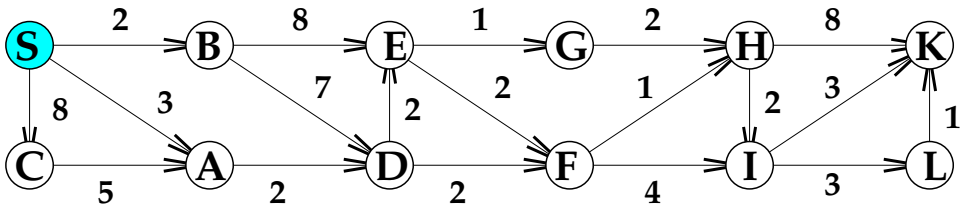
(c) The second subprogram uses dynamic programming, and stores values in an array. What is the asymptotic time complexity of that computation?

(d) What is the asymtotic time complexity of a computation of f(n) using memoization? (Hint: it's a polylograthimic function of n) ---

10. Write the prefix expression equivalent to the infix epression $-a * b - (-c - d) \wedge e$
(Don't forget that $\wedge$ means exponentiation.)

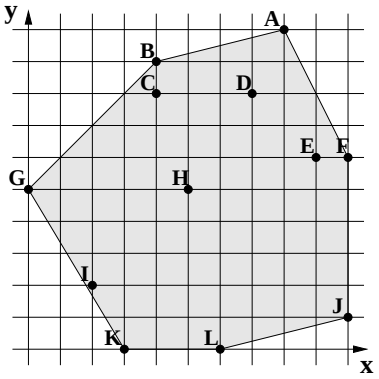
11. Walk through the stack algorithm to change the infix expression $-a + b \wedge c \wedge -f$ to postfix. Show the stack at each step.

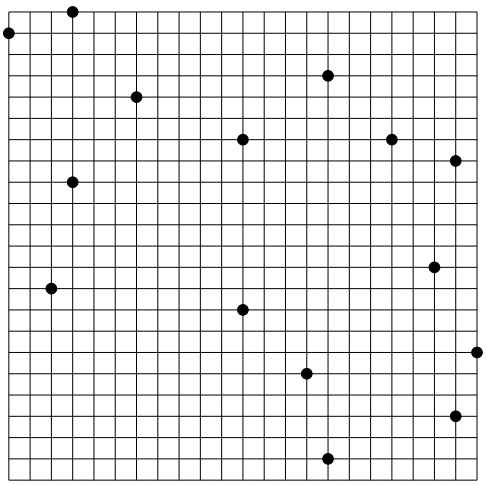
12. Walk through Dijkstra's algorithm for the following graph.



13. The convex hull of a set of a finite set of points in a plane is the smallest convex polygon which encloses the points, together with its interior. Walk through Graham Scan to find the convex hull of the points in the plane given in this figure. (I have not gone over Graham Scan in class yet.)

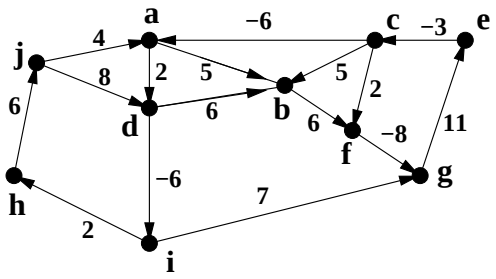
Here is an example, showing the convex hull of the set {A,B,C,D,E,F,G,H,I,J,K}.



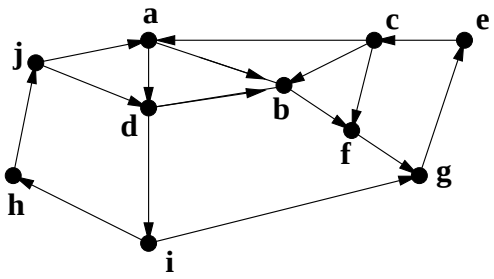


14. Figure (a) below shows an instance of the all-pairs minpath problem. Work the first part of Johnson's algorithm on that graph, showing the adjusted weights in Figure (b).

    Do not complete the computation of Johnson's algorithm.



(a)



(b)

15. True or False.

    (a) _____ If there are 100 data items and 200 possible hash values, a collision is so unlikely that you can, in practice, assume that it won't happen.

    (b) _____ Open hashing uses open addressing.

    (c) _____ Open hashing uses probe sequences.

    (d) _____ You can avoid collisions in a hash table by making the table twice as large as the data set.

    (e) _____ False overflow for a queue can be avoided by implementing the queue as a circular list.

    (f) _____ If a stack is implemented as a linked list, the head of the linked list should hold the top item of the stack.

    (a) _____ Kruskal's algorithm uses dynamic programming.

    (b) _____ There will be no collisions if the size of a hash table is at least the square of the number of data items.

16. Solve each recurrence, expressing each answer in terms of O, Ω, or Θ, whichever is most appropriate.

    (a) $F(n) = F(n/3) + F(2n/3) + 1$

    (b) $G(n) = 2G(n/4) + \sqrt{n}$

    (c) $H(n) = \log n + 1$

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

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

    (f) Solve the recurrence: $F(n) = F(n - \log n) + \log^2 n$

17. Find the time complexity of each of these code fragments in terms of n, using Θ notation.

    (a)
    ```
    for(int i = n; i > 1; i = i/2)
      cout < "Hello world!";
    ```

    (b)
    ```
    for(int i = 2; i < n; i = i*i)
      cout < "Hello world!";
    ```

    (c)
    ```
    for(int i = 1; i < n; i++)
      for(int j = i; j < n; j=2*j)
       cout < "Hello world!";
    ```

18. The asymptotic complexity of the Floyd/Warshall algorithm is _____.

19. The asymptotic complexity of Dijkstra's algorithm algorithm is _____.

20. Here is another coin-row problem. You have a row of coins of various values, where the value of the i^{th} coin is $V[i] > 0$. Write pseudocode which finds the maximum value of a subset of coins, where the set may not contain coins which are either adjacent or just one apart in the row. That is, if the set contains the i^{th} coin, it may not contain either the $(i+1)^{\text{st}}$ coin or the $(i+2)^{\text{nd}}$ coin. For example, if the coins are ⓐ ⓑ ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ in that order, the subset may be {ⓐ, ⓓ, ⓗ}, but not {ⓑ, ⓓ, ⓖ}.

21. Execute heapsort for the list BXQVRST. Show the array at each step, and identify the step at which the array is a heap for the first time.

B	X	Q	V	R	S	T

22. _____ algorithm finds a binary code so that the code for one symbol is never a prefix of the code for another symbol.

23. An acyclic directed graph with 9 vertices must have at least _____ strong components. (Must be exact answer.)

24. In _____ _____ there can be any number of items at a given index of the hash table.

25. You need to store the items A, B, and C, in that order, in a treap. The priority for A is 13, for B is 8, and for C is 14. Use maxheap order. Draw the resulting treap after each insertion, and show each rotation.

26. Consider the function F computed by the recursive code given below.

(a) What is the asymptotic complexity of $F(n)$?

(b) What is the asymptotic time complexity of the recursive code when it computes $F(n)$?

(c) What is the asymptotic time complexity of a memoization algorithm which computes $F(n)$?

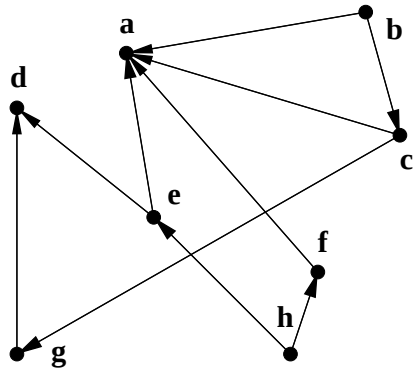
```
int F(int n)
 {
  if(n < 3) return 1;
  else return F(n/3)+2*F((n+1)/3)+n*n;
 }
```

27. If the array $A[5][7]$ is stored in column-major order, how many predecessors does $A[3][4]$ have?

28. You are implementing a 3D triangular array A where $A[i][j][k]$ is defined for $i \geq j \geq k \geq 0$, as a one-dimensional subarray of main memory, and you wish to store A in row-major order, with base address 1024. where $A[i][j][k]$ is defined for $i \geq j \geq k \geq 0$, Each term of A takes one place in main memory. What would be the address, in main memory, of $A[7][4][3]$?

    I have not covered multidimensional triangular arrays. I will postpone that discussion until after the third examination.

29. Write the array of in-neighbor lists and the array of out-neighbor lists for the directed graph shown below.



30. Consider the following recursive C++ function.

```
int f(int n)
 {
  if(n > 0) return f(n/2)+f(n/4)+f(n/4 + 1)+n;
  else return 0;
 }
```

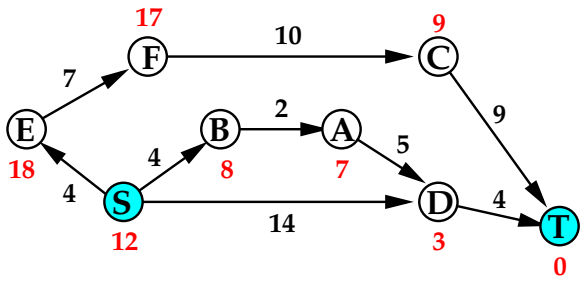
(a) What is the asymptotic complexity of f as a function of n, using Θ notation?

(b) What is the asymptotic time complexity of this code as a function of n, using Θ notation?

(c) Write pseudo-code for a dynamic programming algorithm to compute $f(n)$ for a given n. What is the asymptotic time complexity of your code as a function of n, using Θ notation?

(d) Write pseudo-code for a memoization algorithm to compute $f(n)$ for a given n. What is the asymptotic complexity of the algorithm in terms of n, using Θ notation?

31. Walk through the A^* algorithm for the weighted directed graph shown below, where the pair is (S, T). The heuristic is shown as red numerals.



.

Show the arrays and the contents of the heap at each step. h is the heuristic, g is the current distance from the source, f is the sum of h and g. Label processed vertices with g and f, and show the backpointers.

32. Find the Levenshtein edit distance from the word "mennoover" to the word "maneuver." Show the matrix.