University of Nevada, Las Vegas Computer Science 477/677 Fall 2024

Assignment 6: Due Saturday November 9, 2024

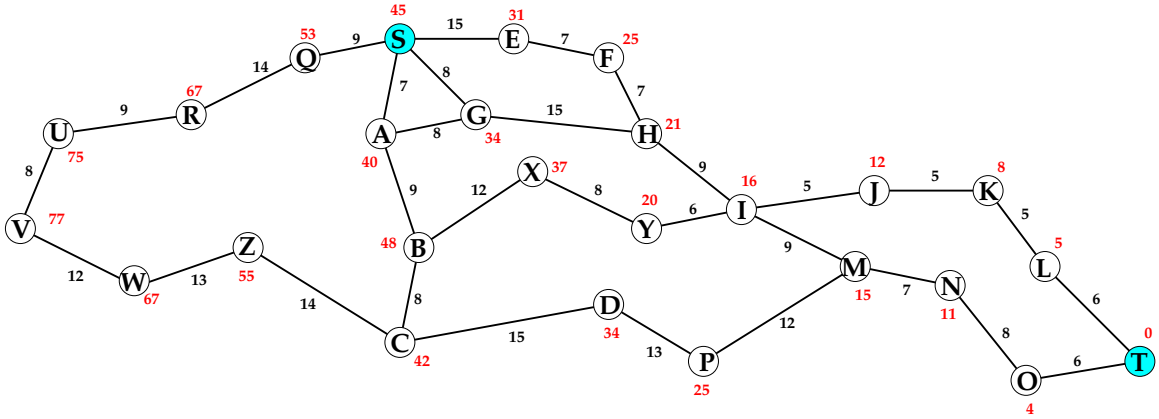
Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet.
To turn in the homework, follow instructions given by the graduate assistant, Sepideh Farivar. at `farivar@unlv.nevada.edu`.

AI Overview, from the internet: "In the context of hash tables, 'open hashing' refers to a collision resolution strategy where colliding elements are stored in separate linked lists outside the main hash table array, while 'closed hashing' (also called open addressing) stores colliding elements within the array itself by probing for an available slot when a collision occurs; essentially, open hashing uses external data structures to handle collisions, while closed hashing uses the same array to store all elements, even if they collide."

1. Solve these recurrences, expressing the answers using Θ notation.
   (a) $F(n) = F(n/2) + 2F(n/4) + n$
   (b) $F(n) = 2F(2n/3) + F(n/3) + n$
   (c) $F(n) = 3F(n/2) + 3F(n/4) + n^2$
   (d) $T(n) = T(7n/10) + T(n/5) + n$

2. Walk throuth the A^* algorithm for the following weighted graph, finding the least cost path from S to T. The edge weights are in black and the heuristics are in red. The heuristics are both admissible and consistent. Your answer should label each fully processed vertex with both f and g values. not all vertices will be processed.



3. The following recursive C++ program computes a function `george(n)` for a given integer n.

```
int george(int n)
 {
  if(n <= 1} return 1;
  else
   return george(n/2) + george(n/3) + george(n/6) + n;
 }
```

You only want to compute george(n) for some n.

(a) _____ What is the asymptotic complexity, in terms of n, of the value of george(n)?

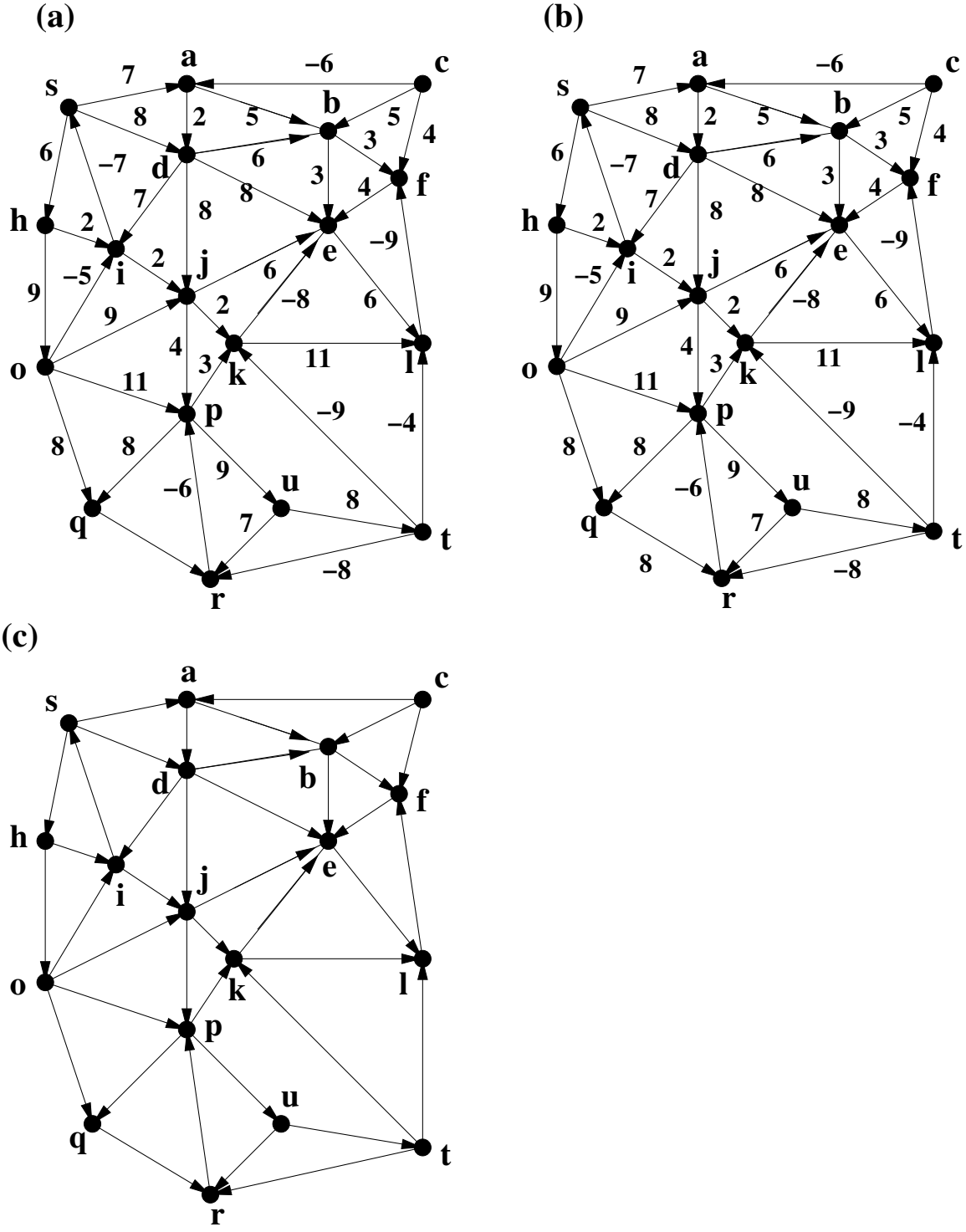
(b) _____ What is the asymptotic complexity, in terms of n, of the time required to vind george(n) using the recursive program given above? (Hint: the answer is not the same.)

(c) _____ What is the asymptotic complexity, in terms of n, of the time required to compute george(n) using dynamic programming? That means, computing george(i) for all i from 0 to n in order.

(d) _____ What is the asymptotic complexity, in terms of n, of the time required to compute george(n) using memoization? (Hint: how many intermediate values will be stored?)

4. Figure (a) below illustrates a weighted directed graph. Figures (b) and (c) are copies. In Figure (c), show the adusted arc weights needed to work Johnson's algorithm. Use (b) for your intermediate work, showing the non-positive values at the vertices. To avoid clutter, do not draw the zero-weight arcs from s to the other vertices.

   Do not finish Johnson's algorithm.



5. Evaluate

(a) $\dfrac{\log_5 8}{\log_5 4}$

(b) $\dfrac{\log_3 2}{\log_3 \left(\frac{1}{2}\right)}$

(c) $\log_5 9 \cdot \log_3 5$

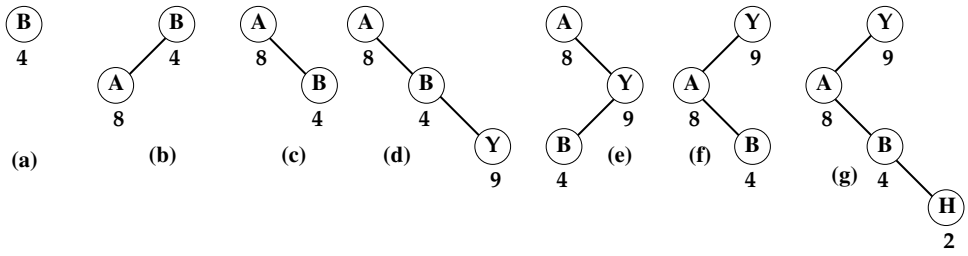
(d) $\log_2 \sqrt{8}$

(e) $2^{\log_2 9 - \log_2 3}$

(f) $4^{\log_2 5}$

6. Insert the following items into a treap, in the order given. The priority of each item is given. Show all the insertions and rotations. Your treap should use max-heap order. Your first seven steps are shown below. In total, there should be 15 (or so) figures.

B	4
A	8
Y	9
H	2
F	5
D	10



Hashing

Here is a link to notes from a class on hashing at Carnegie Mellon University.
`https://www.cs.cmu.edu/~avrim/451f11/lectures/lect1004.pdf`
And here are notes from a class on hashing at the Univerity of Washington.
`https://courses.cs.washington.edu/courses/cse326/06su/lectures/lecture11.pdf`

7. Suppose we with to store 3-letter names in a hash table. As in the CMU page,, each name is converted into a binary string by replacing each of letter with a binary string of length 5, as given in the table below. For simplicity, our code is case-insensitive, so "Bob" is written "000100111100010"

a	0	0	0	0	1	n	0	1	1	1	0
b	0	0	0	1	0	o	0	1	1	1	1
c	0	0	0	1	1	p	1	0	0	0	0
d	0	0	1	0	0	q	1	0	0	0	1
e	0	0	1	0	1	r	1	0	0	1	0
f	0	0	1	1	0	s	1	0	0	1	1
g	0	0	1	1	1	t	1	0	1	0	0
h	0	1	0	0	0	u	1	0	1	0	1
i	0	1	0	0	1	v	1	0	1	1	0
j	0	1	0	1	0	w	1	0	1	1	1
k	0	1	0	1	1	x	1	1	0	0	0
l	0	1	1	0	0	y	1	1	0	0	1
m	0	1	1	0	1	z	1	1	0	1	0

We are letting $M = 8 = 2^3$, that is, $M = 3$, and each datum is encoded as a bit string of length 15. Therefore, the hash function h is defined by a 3×15 matrix of Boolean values. I have chosen those values at random.

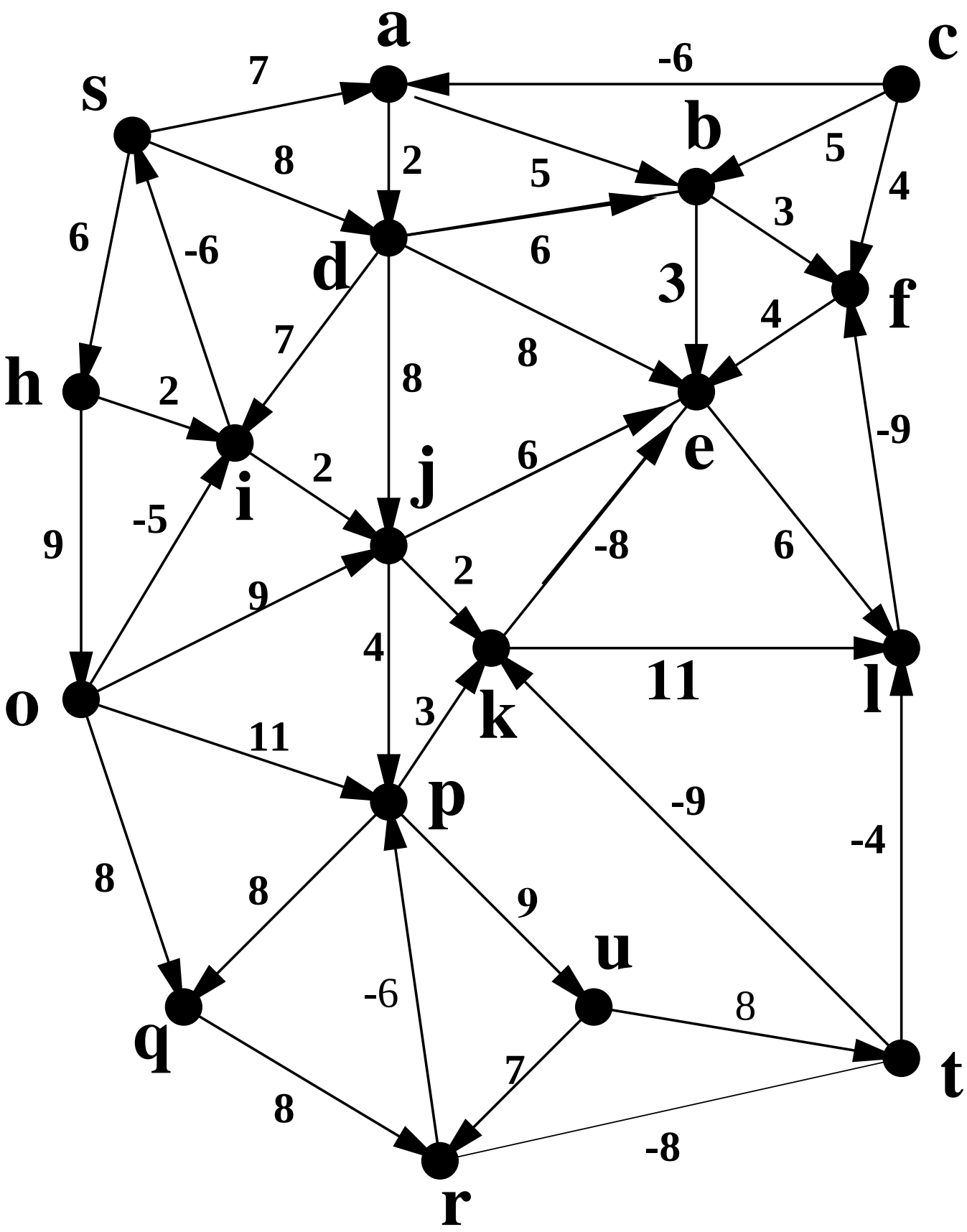
1	1	0	1	0	0	0	0	0	1	1	0	1	0	1
0	1	0	0	1	0	0	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	0	1	0	1	1	1	0

Following the instructions on the CMU page, We compute the hash value $h(\text{Bob}) = 3$.

We give the following list of $N = 6$ data:

Bob
Ann
Sue
Ted
Van
Liz

Compute the hash value of each of the data. How many collisions should we expect? How many collisions are there actually?


a
c
s
7
-6
b
8
2
5
5
4
6
3
-6
d
6
3
f
4
7
8
h
8
2
e
-9
i
2
j
6
-5
9
-8
6
2
9
4
11
l
o
3
k
11
p
-9
-4
8
8
9
u
-6
8
q
7
t
8
-8
r