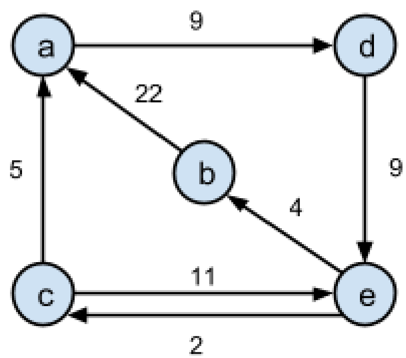
Shuni Li
Comp221 HW1
Professor Shilad Sen
Feb 8, 2016

1. (10 pts) Consider the picture of the weighted, directed graph below. Show the adjacency matrix and adjacency list representations for this graph.



$$
\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} &
\begin{bmatrix}
\infty & \infty & \infty & 9 & \infty \\
22 & \infty & \infty & \infty & \infty \\
5 & \infty & \infty & \infty & 11 \\
\infty & \infty & \infty & \infty & 9 \\
\infty & 4 & 2 & \infty & \infty
\end{bmatrix}
\end{array}
$$

a → d,9
b → a,22
c → a,5 → e,11
d → e,9
e → b,4 → c,2

FIGURE 1 (a) Adjacency matrix. (b) Adjacency lists.

2. (10 pts) For each of the following applications, indicate the most appropriate data structure.
   a. Answering telephone calls in the order of their known priorities.
   b. Sending backlog orders to customers in the order they have been received.
   c. Implementing a calculator for computing simple arithmetic expressions.

   a. Priority Queue; It allows every element to have a priority rank. And an element with high priority is served before an element with low priority.
   b. Queue; It allows the first element in queue to be processed first, which resonates the process of sending back orders.
   c. Stack; every finished operation should be pushed on the stack and waiting for further operations.

3. (10 pts) What would be printed by invoking the algorithm below with an input of 21?

```
Algorithm MysteriousRecursion(num)
    if num <= 0 then
        print "Done"
    else
        print num
        MysteriousRecursion(num - 2)
```

The result will be:

21 19 17 15 13 11 9 7 5 3 1 Done

4. (20 pts) Decide whether each of the following are true or false. Justify your answer with a good proof or informal argument.

   a. $n(log_2 n^2) + 15n \in O(n^{2.15})$

      True.

      We start with simplifying the expression: $n(log_2 n^2) + 15n = 2n(log_2 n) + 15n$. When n is positive and large, $(log_2 n)$ will be greater than 1. So the growth rate of $n(log_2 n)$ will be much greater than n. Since we always take the term with largest growth rate and take off constant coefficients, this is a $O(n\ log_2 n)$ algorithm, which is much smaller than $n^{2.15}$. Thus, it is true.

   b. $n(log_2 n^3) + 15n \in \Theta(n^3)$

      False.

      We start with simplifying the expression: $n(log_2 n^3) + 15n = 3n(log_2 n) + 15n$. When n is positive and large, $(log_2 n)$ will be large and the growth rate of $n(log_2 n^2)$ will be much greater than n. Since we always take the term with largest growth rate and take off constant coefficients, this is a $\Theta(n\ log_2 n)$ algorithm, which is much smaller than n^3.

   c. $2^{10} n 2^{n+1} \in \Theta(2^n)$

      False

      Simplifying the expression: $2^{10} n 2^{n+1} = 2^{10} n * 2 * 2^n = 2^{11} n 2^n$

      After dropping the constant coefficients, we have $\Theta(n2^n) \notin \Theta(2^n)$.

   d. $\frac{1}{100} n^2 + 831n - 22 \in \Omega(n^2)$

      True

      n^2 has a much larger growth rate than n. After taking the largest term and dropping constants, we have $\Omega(n^2)$. So it is true