COM S 311

Homework 5

Recitation 5, 1-2pm, Marios Tsekitsidis

Christian Shinkle

April 26, 2018

1. If we place every vertices in V into a priority queue, back by a max-hep, and pritortize them based on the size of their degree. We can reduce the runtime of this algorithm $O(nlogn)$ where n is the number of vertices in V.

2. (a) The algorithm is as follows:
   Input: Graph G and Set S
   Let $visited$ be an array of size $|S|$ where all elements are 0.
   for each $s \in S$
   for each e that is adjacent to s
   if $visited[e] == 0$
   $visited[e] \leftarrow 1$
   else
   return false
   return true
   The largest S can be is $|V| = n$, so it will take $O(n)$. The outer loop will take n time and the inner loop will take n time, so the loops will take $O(n^2)$ time. Therefore the runtime is $O(n) + O(n^2) \in O(n^2)$.

   (b) The algorithm is as follows:
   Input: Graph G
   Calculate all 2^n combinations of independent sets
   Let S be the largest of all the sets
   return S
   This problem is NP, so the runtime will be at least in 2^N. It will take n time to calculate a single independent set and there are 2^n number of sets possible in the worst case, so this algorithm is $\in O(2^n)$.

3. The algorithm is as follows:
   Input: $n \times m$ matrix M
   Let C be an $n \times m$ matrix that has all cells initialized to $-\infty$.
   for i in the range 1 to n
   $C[i, 1] \leftarrow M[i, 1]$

for i in the range 1 to n
    for j in the range 2 to m
        if $i-1$ is in bounds and $M[i-1,j-1]+M[i,j] > C[i,j]$
            $C[i,j] \leftarrow M[i-1,j-1]+M[i,j]$
        if $M[i,j-1]+M[i,j] > C[i,j]$
            $C[i,j] \leftarrow M[i-1,j-1]+M[i,j]$
        if $i+1$ is in bounds and $M[i+1,j-1]+M[i,j] > C[i,j]$
            $C[i,j] \leftarrow M[i+1,j-1]+M[i,j]$
return the max value in the m column of C.
The recurrence relationship is as follows:
$T(i,j) = \max(T(i-1,j-1),\ T(i,j-1),\ T(i+1,j-1)) + M[i,j]$
The runtime of the iterative algorithm is as follows:
The big-O of initailizing C is $O(nm)$ because there are $n \times m$ cells in C and every cell will be visisted. The big-O of the first loop is n trivially. The big-O of the nested for loops is $O(nm)$ because every cell will be visited 4 times and the visiting of the cell will take a constant number of opertation. Lastly, the final loop will take $O(n)$ trivially. Therfore, the runtime of the algorithm is $cnm + cn + cnm + cn \in O(nm)$.

4. The algorithm is as follows:
Input : set S of n non-negative integers
Let N be the sum of all $x_i \in S$.
if N is odd, then return false and terminate
Let N' be 0.
Let $best \leftarrow N$
While $best \neq 0$
    Let min be ∞.
    Let $index$ be 1.
    for i in the range 1 to n
        if $|best - 2 * S[i]| < min$
            $min \leftarrow |best - 2 * S[i]|$
            $index \leftarrow i$
    if $min < best$
        $N \leftarrow N - S[index]$
        $N' \leftarrow N' + S[index]$
        $best \leftarrow min$
        remove $S[index]$ from S
    else
        return false and terminate
return true

The runtime of the iterative algorithm is as follows:
Calculating the sume of all $x_i \in S$ will take N time. The while loop will run at most n times. The for loop will run at most n times. Therefore, the runtime is $O(N + n^2)$.

The recurrence relationship is as follows:
$T(n) = T(n-1) + n$.