Homework 1
Colorado CSCI 5454

Sai Siddhi Akhilesh Appala

November 30, 2022

People I studied with for this homework: None

Problem 1

Part a

$$
\begin{aligned}
f(x) &= 5n^3 + 2n \\
5n^3 \leq 5n^3 + 2n &\leq 5n^3 + 2n^3 \\
5n^3 \leq f(n) &\leq 7n^3 \\
g(n) &= n^3 \\
5.g(n) \leq f(n) &\leq 7g(n) \\
c1 = 5, c2 = 7, g(n) = n^3, N = 1, \forall n \geq N &
\end{aligned}
$$

the above values are satisfying the given condition

Part b

Solution 1: Equation to be proved – $f(n) < c.g(n)$ using hint of part a

$$
\begin{aligned}
f(x) &= 5n^3 + 2n \\
5n^3 + 2n &< 5n^6 + 2n^6 \\
f(n) &< 7n^6 \\
g(n) &= n^6 \\
f(n) &< 7g(n) \\
c &= 7 \\
g(n) &= n^6
\end{aligned}
$$

Solution 2: using limits tends to 0
Need to be proved $\boldsymbol{f(n) \in o(n^6)}$

$g(n) = n^6$;

need to prove this statement $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$\lim_{n\to\infty} \frac{5n^3+2n}{n^6}$

$\lim_{n\to\infty}(\frac{5}{n^3} + \frac{2}{n^5})$

=0+0=0

Part c

solution 1 using tight bound

equation to be proved – $f(n) > c.g(n)$

$$
\begin{aligned}
f(x) &= 5n^3 + 2n \\
5n^3 + 2n &> 2n \\
f(n) &> 2n \\
g(n) &= n \\
f(n) &> 2g(n) \\
c = 2, \forall n &\geq N \\
g(n) &= n
\end{aligned}
$$

Problem 2

Solution 1

$f(x) = \sum_{j=1}^{n} \sqrt{j}$

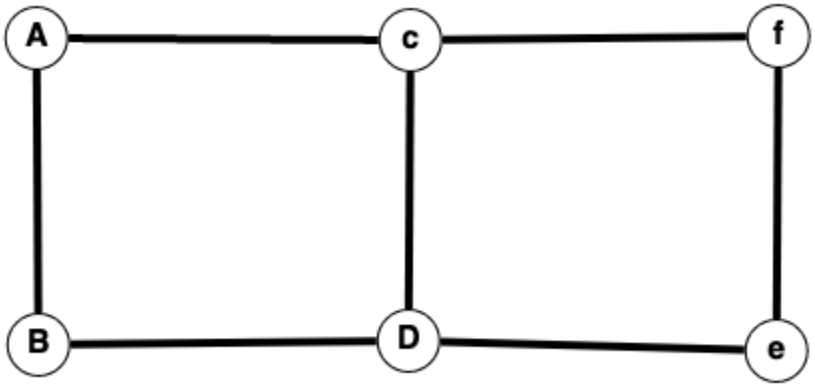
$\sqrt{n/2} + ... + \sqrt{n} \leq \sqrt{1} + \sqrt{2} + \sqrt{3} + ... + \sqrt{n} \leq \sqrt{n} + \sqrt{n} + \sqrt{n} + ... + \sqrt{n}$

$n/2 * \sqrt{n/2} \leq f(x) \leq n * \sqrt{n}$

$1/(2\sqrt{2}) * n^(3/2) \leq f(x) \leq n^(3/2)$

From above -$c1 = 1/(2\sqrt{2}), c2 = 1, N = 1, g(n) = n^(3/2)$, $\forall n \geq N$

Problem 3

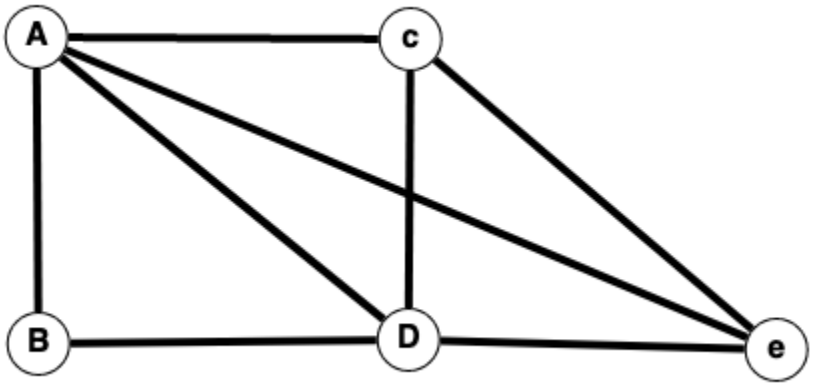


The Claim is false. Please refer to the below example
The longest simple path from B to e can be B, A, c, f, e but we can also go with a B, D, e. Which is not satisfying the claim.

Problem 4

Explanation :
A directed acyclic graph (DAG) is a directed graph with no directed cycles. That is, it consists of vertices and edges, with each edge directed from one vertex to another, such that following those directions will never form a closed loop.

Before: A undirectional graph with vertices



Algorithm Explanation:

1. Use DFS(G) for traversing an un-directional graph.
2. While traversing check whether its neighbors are visited or not
3. if Visited do not assign any direction to the edge.
4. else assign an **outside** direction to the edge starting from the vertex.
5. Complete the DFS(G) traversal for each vertex and assign the direction if un-visited.

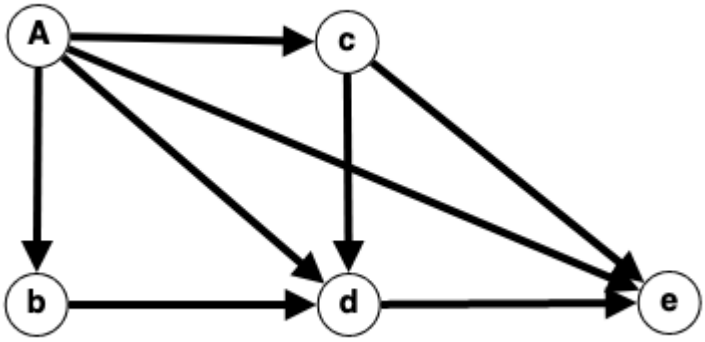
Algorithm 1 DFS traversal for un-directional graph

```
Input: Graph G = (V, E)
Define array is marked of length n
Set is marked[v] = False for v = 1, . . . , n
Call explore-unvisited-graph(s)
Return list T
```

Subroutine 2 explore-unvisited-graph(v)

```
push vertex v to the list T
Set is_marked[u] = True
for each neighbor u of v do
   if is_marked[u] == false then
      is_marked[u] = true
      push vertex v to the list T
      Assign an outward directed edge from v to u(v → u) by updating G.
   end if
end for
explore-unvisited-graph(u)
return
```

After: A graph with the directions and having a topological sort



Proof - If outward edge u $\longrightarrow$ v, U is before V in the list

1. From the above algorithm we can see that while adding a direction to the edge U$\rightarrow$ v which is called in the [explore-unvisited-graph(u)]subroutine of U. the edge will only be assigned if the U is unvisited previously and V is unvisited previously.

2. we immediately add U to the list as it is called in the subroutine of U. and then V will be added later in the list while giving directions to the neighbors of V.

3. This indicates that v is always after u in the list. which is satisfying and providing the topological sorting order.