Problem 1.

Let $G$ be a DAG where the vertices are labelled with numerical values.

A. Write a function $\text{MaxReachable}(u)$ which returns the maximum label on a vertex reachable from vertex $u$ (including $u$ itself.)

B. Write a function $\text{TotalReachable}(u)$ which returns the sum of the labels on vertices reachable from vertex $u$.

C. Write a function $\text{MaxPathFrom}(u)$ which returns the maximum sum of the labels on any path starting at $u$.

All these should run in time linear in the size of $G$.

For example, in the graph below:

$\text{MaxReachable}(B) = 10$ (corresponding to F).
$\text{TotalReachable}(B) = 35$ ($B+C+D+E+F+G+H$)
$\text{MaxPathFrom}(B) = 27$ (corresponding to $B-D-C-E-G$).
Problem 2

Trace the execution of Dijkstra’s algorithm on the following graph, taking A as the starting vertex. Show the successive states of the array $D[i]$ and of the set of vertices whose distance has been determined.

![Graph](image)

Problem 3

(Siegel Ex. 7.21). Write an algorithm that takes a DAG $G$ as input and prints out all the possible topological sorts of $G$. For instance, given the graph below, the algorithm would output

- $A, B, C, D, E$
- $A, B, C, E, D$
- $A, B, D, C, E$
- $A, C, B, D, E$
- $A, C, B, E, D$

![Graph](image)

It should print out each sort only once. Your algorithm does not have to produce the sorts in this order.