

Deadline: April 17th

Problem: Find a vertex cover of a given weighted graph, using the primal dual algorithm seen in class.

Input: Number of vertices (n), list of vertex weights ($w[1]$ to $w[n]$), adjacency lists of each vertex.

Output: Sequence of tight edges, sequence of vertices as they are picked into the vertex cover, and the cost of the vertex cover.

Example 1:

Sample Input:

n : 4

Weights: 5, 6, 10, 1

Adj lists:

1:2,3

2:3,1

3:1,2,4

4:3

Note that different outputs are possible depending on the order in which the dual variables corresponding to edges are processed.

Sample output 1 :

Edge sequence: (1,3), (2,3)

Vertex cover: 1,3

Cost: 15

Sample output 2:

Edge sequence: (2,3), (3,4), (1,3)

Vertex cover: 2,4,3

Cost: 17

Example 2:

Sample Input:

n: 6

Weights: 3,5,4,1,5,6

Adj lists:

1:2,3

2:3,1,4

3:1,2,5

4:2,5,6

5:3,1,6

6:4,5

Sample output 1 :

Edge sequence: (1,2), (2,3),(3,5),(4,6)

Vertex cover: 1,2,3,5,4

Cost: 15

Sample output 2:

Edge sequence: (5,6), (4,6), (2,3),(1,2)

Vertex cover: 5,4,3,2

Cost: 12