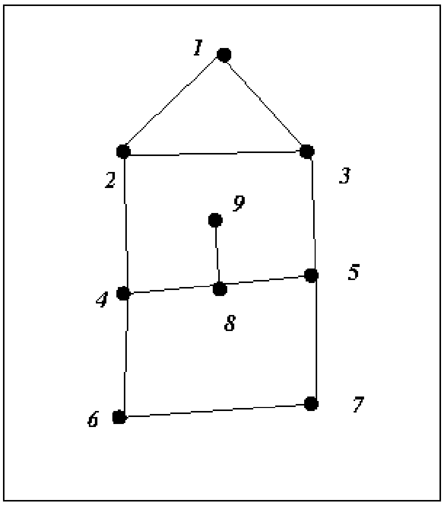
End-Semester Exam Solutions
CS 213
2021
Marks=3+6+8+8+6+6=37

Instructions:
1. **Make reasonable assumptions. If in doubt, send a message to the proctor. Cribs about assumptions will not be entertained later.**
2. **Mark distribution in sub-parts of the problem are indicative. They may change based on earlier parts.**

1. [3 marks] Recall the function mysearch(s) from the Tutorial 9 to search a graph, starting at vertex s. It uses a set SS and two functions add(v,SS), which adds a vertex and select(SS) which returns some element of SS.

```
Function mysearch(s)
Global visited;
For all u visited[u]=false; visited[s]=true;

SS=empty;
add(s,SS); nos=1; record[nos]=s;
While nonempty(SS) // search begins
   y=select(SS);
   Print(y); // print vertex
   nos=nos+1; record[nos]=y;
   For all v adjacent to y // process y
     If visited[v]==false
        visited[v]=true;
        add(v,SS);
     Endif;
   Endfor; // done processing y
   Print(SS); // print set
Endwhile // search ends
Endfunction;
```



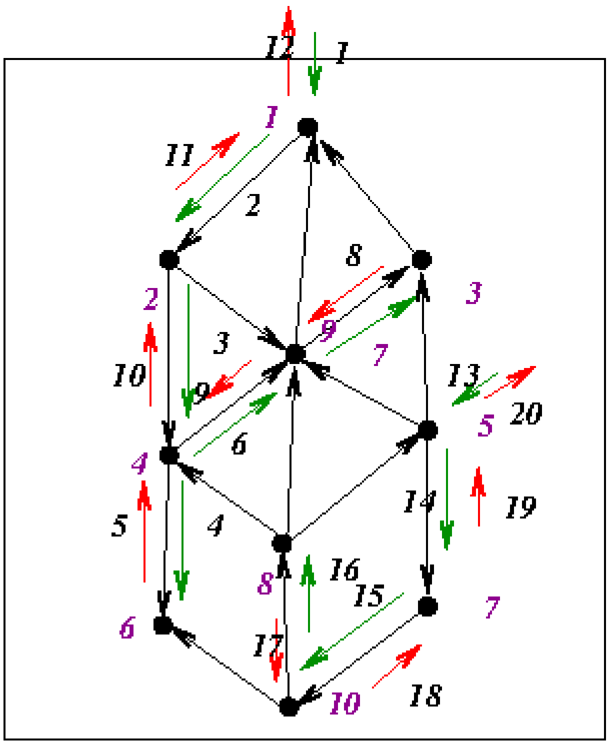
The above mysearch is run on the graph above starting at vertex 1. Note that there are two print statements. Suppose that the values of Print(y) are as given below. Write the output of Print(SS).

nos	Print(y)	Print(SS)
1	1	2,3
2	2	3,4
3	4	3,6,8
4	8	3,6,5,9
5	3	6,5,9
6	5	6,9,7
7	9	6,7
8	6	7
9	7	-

2. [6 marks] Recall the DFS for directed graphs which begins at a specified vertex v and explores all vertices reachable from that vertex, then picks the lexicographically next unvisited vertex to explore and so on, as in Tutorial 10. For the graph below run DFS starting at vertex 1. Based on that, fill in the in[w] and out[w] times for each vertex in the table below. Let mat[w] of a vertex w be the minimum of arrival times of all edges going out from the subtree rooted at v. Fill that the table below. [4 marks]

vertex	DFS starting at 1

	in[v]	out[v]	mast[v]
1	1	12	X
2	2	11	1
3	7	8	1
4	3	10	1
5	13	20	3
6	4	5	X
7	14	19	3
8	16	17	3
9	6	9	1
10	15	18	3



[2 marks] What are the connected components and what are the edges between them?

The strongly connected components are A=[1,2,4,3,9], B=[6], C=[5,7,8,10]. The edges are A->B, C->B, C->A.

3. [8 marks] An explorer wants to cross a desert in a Jeep. Her journey is from point A0 to another point A5 across the desert, 5 days away. But the Jeep can carry fuel for

only 4 days of journey. However, the explorer can create fuel dumps at any or all of four intermediate points A1, A2, A3 and A4 at a distance of 1,2,3 and 4 days from A0. There are 20 days of fuel available at A0. Model this problem of finding a plan for the journey as a graph problem.

(A) [2 marks] Describe the vertices and their **precise** interpretation.

The vertices are 6-tuples: v=[n0,n1,n2,n3,n4,n5] where n0+n1+n2+..+n5<=20. They represent the fuel available on the morning of a particular day. In addition, the location of the explorer on the morning of that day is also marked. Thus the vertex set is (v,i) where v is the fuel available and i is the location of the explorer, i=0…..5.

(B) [2 mark] Describe the edges.

There is an edge from (v,i)->(w,j) if (i) j=i-1 or i+1. w(i)=v(i)-k and w(j)=v(j)+k-1. For example
([10,4,1,0,0,0],2) -> ([10.4,0,0,0,0],1) is an edge which says that the explorer went from A2 to A1 using up 1 unit of fuel available at A2.

(C) [2 marks] What is the starting node **s** and the ending node **t**?

([20,0,0,0,0,0],0) is the unique starting node. Any node (v,5) is the ending node.

(D) [2 mark] Show any path of length 3 in your graph starting at **s**, and explain in 3 sentences what steps of the plan it models.

([20,0,0,0,0,0],0)->([16,3,0,0,0,0],1)->([16,2,0,0,0,0],0)->([12,5,0,0,0,0],1)

Day 1: Start at A0 with 4 units. Burn 1 unit, keep 3 units. Reach A1
Day 2: Start with 1 unit and reach A0. Keep 2 units of fuel at A1.
Day 3: Start from A0 with another 4 units and reach A1 with 5 units of fuel.

4. [8 marks] We are given a transportation network given as directed graph G(V,E) with s as a designated start node. We are also given a fare function f(u,v) on edges, where f(u,v) is the fare on the edge (u,v). We would like to compute shortest fare paths from s to every other vertex v under the following extra condition. We have been given one coupon which allows for one free journey on any edge of our choice.

(A) [3 marks] State the basic idea to solve the problem and why it is correct, in 3 precise sentences.

The basic idea is to create two copies of the graph G0 and G1 with the same fares, where G0 tracks movement without coupon and G1, with coupon. For every edge (u,v) in G, we create an edge (u0,v1) from vertex u in G0 to vertex v in G1 of fare 0. Thus you transit from u to v without paying a fare but from

G0 to G1. Thus every path in the new graph is a path which stays in G0 or transits exactly once to G1 and pays the same fares, as required.

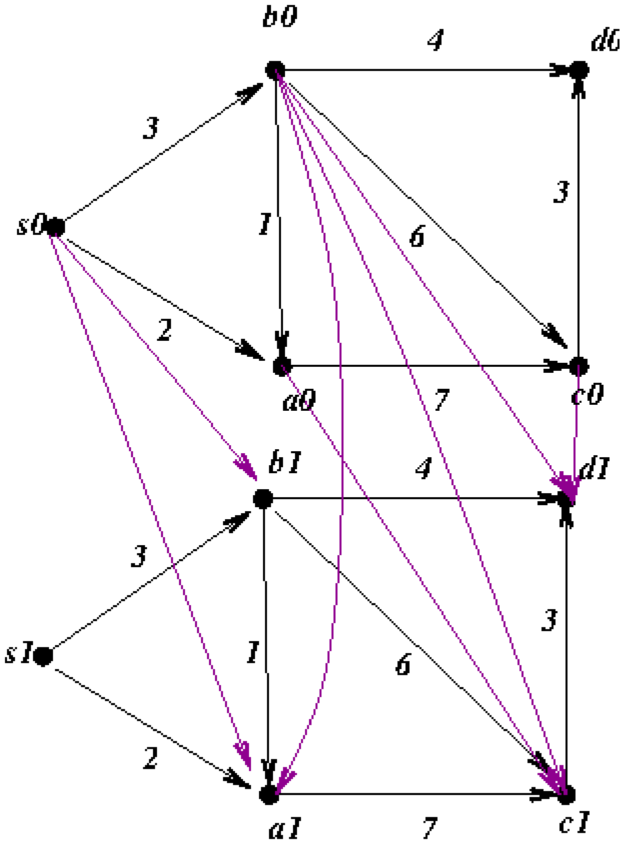
(B) [2 marks] Write down in clear handwriting and in ONE page, the pseudocode with comments, for the above.

```
Function coupondisjstra.

V0=copy(V); V1=copy(V); E0=copy(E); E1=copy(E); f0=copy(f); f1=copy(f)
VV=V0 union V1, EE = E0 union E1, ff=f0 union f1;
For all (u,v) in E
    e=(u0,v1)
    Append e to EE
End;
dd=Dijkstra(GG,ff);
For all v in V
  d(v)=dd(v1);
End;

Endfunction;
```

(C) [3 marks] Run the pseudocode in tabular form for the graph below.



s0	a0	b0	c0	d0	a1	b1	c1	d1	Queue	u=del min	d(u)
0	i	I	I	I	I	I	I	I	s0	s0	0
-	2	3	I	I	0	0	I	I	a0, b0, a1, b1	a1	0
-	2	3	I	I	-	0	7	I	a0, b0, b1, c1	b1	0
-	2	3	I	I	-	-	7	4	a0, b0, c1, d1	a0	2
-	-	3	9	I	-	-	2	4	b0, c0, c1 d1	c1	2
-	-	3	9	I	-	-	-	4	b0, c0, d1	b0	3
-	-	-	9	7	-	-	-	3	c0, d0, d1	d1	3
-	-	-	9	7	-	-	-	-	c0, d0	d0	7
									c0	c0	9

Thus the distances are d(a1)=0, d(b1)=0, d(c1)=2 d(d1)=3

A standard wrong solution is to find the shortest path in the original graph and to use the coupon on the longest edge. This has a maximum of 3 marks out of 8. Another standard solution has been to make one edge as 0 fare and solve m such instances and take the minimum cost edge. This is correct but inefficient and is awarded 4-6 marks.

5. [6 marks] Recall the Huffman code algorithm. Suppose that it was run on an alphabet {a1,a2,a3,a4,a5} with frequencies F={f1,f2,f3,f4,f5} to obtain the tree below with

leaves L1,...,L5 and internal nodes A,B,C,D. Note that the algorithm proceeds by selecting two letters of the minimum frequency, merging them as leaves and forming a new composite letter with a new frequency. Each internal node represents a merge of two current letters.

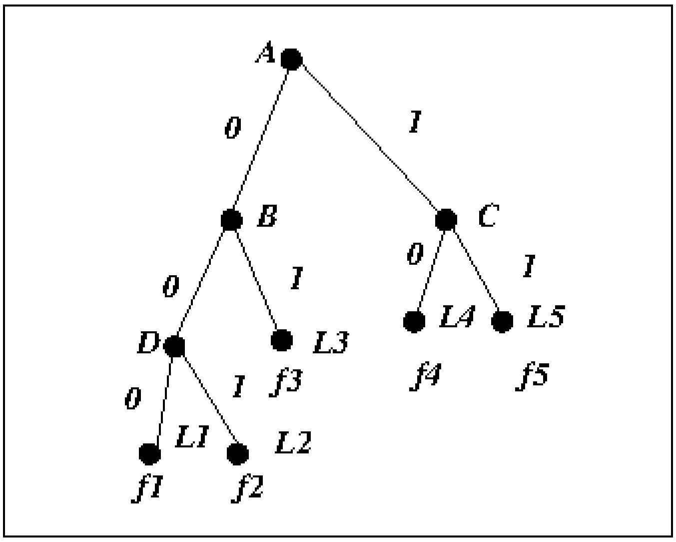
(i) [1 marks] Given the tree below, what are the possible order of merges A,B,C and D. Why (in 2 sentences)?

A must be last because B and C must merge first. B must merge after D. That gives us 3 possibilities in the table below.

(ii) [3 marks] For each order of merges above, write the inequalities in the variables f1,...,f5, which will ensure that the merge order is as chosen.

See the table below.

(iii) [2 marks] Illustrate a frequency vector F which will produce the merges in the chosen order.



A	B	C	D	Inequalities	Feasible vector
4	2	3	1	(f1,f2<=f3,f4,f5)(f1+f2<=f4,f5)(f3<=f4,f5)(f4,f5<=f1+f2+f3)	(1,1,1,2,2)
4	3	2	1	(f1,f2<=f3,f4,f5),(f4,f5<=f1+f2,f3),(f1+f2,f3<=f4+f5)	(1,1,1,1,1)
4	3	1	2	(f4,f5<=f1,f2,f3),(f1,f2<=f3,f4+f5),(f1+f2,f3<=f4+f5)	(1,1,1,1,1)

6. [6 marks] Mr. Gokhale has a bungalow called BlueSky by the beach which he lets out on a daily basis for any number of days. His bungalow is very popular and he gets frequent phone calls asking for availability of the bungalow for k consecutive days.

   (A) Describe the strategy in 3 sentences.
   Update fills up WBS from k=0 to 10, going through Calendar from left to right, and from date+1 to N. It uses next to locate the next bunch of zeros. It matches this bunch against the k which is currently filled and updates k if the size of this bunch is greater than k. If next returns empty, then it fills up the remaining k with N+1.

   (B) Specify and describe the auxiliary functions to be used.

```
[first,last]=next(n,N)
// returns the first  maximal consecutive sequence of zeros from n+1 to N in
// Calendar. first=n if there is no such sequence
```

   (C) Write commented pseudocode and an outline.

```
Function UpdateWBS(date, Calendar, N)

k=0; n=date; seq=1;
While k<10 // WBS is not full
   [first,last]=next(n,N)
   If first>n
        kk=last-first+1;
        kk=min(kk,10) // only upto 10
        For j=k+1:kk
            WBS(j)=first;
         End; // of for
        n=last+1;
       k=kk;
       print(seq);
      seq=seq+1;
  Else
       For j=k+1:10
          WBS(j)=N+1
       End; // of For
       Print("Free");
      k=10;
   End; // of If
End; // while
Endfunction;
```

What is the time complexity of your code in terms of N and the size of WBS (in this case, 10). .

The complexity of update does not depend on “10”. It depends linearly on the number N-n. Note that next is called on increasing indices and never checks a Calendar entry twice.

(D) Execute it for the input calendar with Date 802 and N=819.

	WBS(1)	WBS(2)	WBS(3)	WBS(4)	WBS(5-10)
Date=802 N=819	804	807	807	815	820