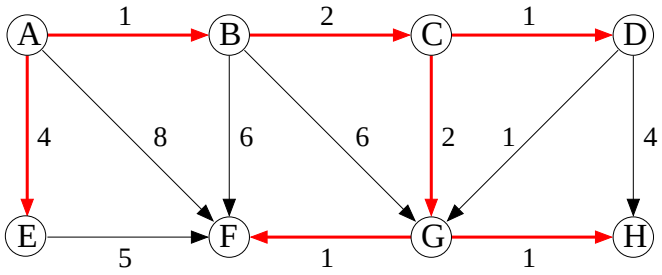
University of Nevada, Las Vegas Las Vegas Computer Science 477/677 Fall 2019

Assignment 5: Due Wednesday October 9, 2019

Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets, but only by stapling. Turn in the pages to the graduate assistant at the beginning of class, September 11.

1. Work Problem 4.1 on page 120 of your textbook.



	0	1	2	3	4	5	6	7	8
A	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
B	∞	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
C	∞	∞	**3**	**3**	**3**	**3**	**3**	**3**	**3**
D	∞	∞	∞	**4**	**4**	**4**	**4**	**4**	**4**
E	∞	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
F	∞	**8**	**7**	**7**	**7**	**7**	**6**	**6**	**6**
G	∞	∞	**7**	**5**	**5**	**5**	**5**	**5**	**5**
H	∞	∞	∞	∞	∞	**8**	**6**	**6**	**6**

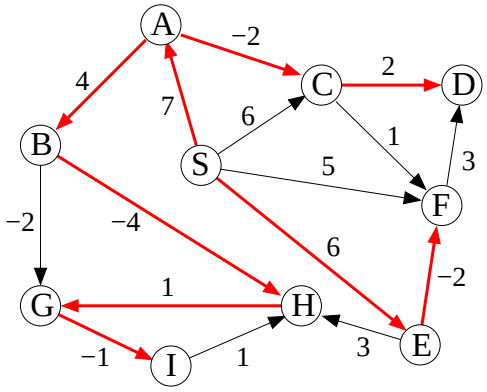
Each entry shows the current best distance from A of that node at that step. The colors of the table encode the data structure. A red numeral indicates that the corresonding node is in the priority queue at that step. numbers may be changed. A black or blue numeral indiates the minimum distance from A to that node. A blue numeral indicates that the corresponding node is deleted from the priority queue at that step, and its subsequents are visited and their The shortest path tree is shown in red.

4. You are working on computer which lacks multipliation and addition. However, it can add or subtract 1 or 2. What does this function do? What is its loop invariant?

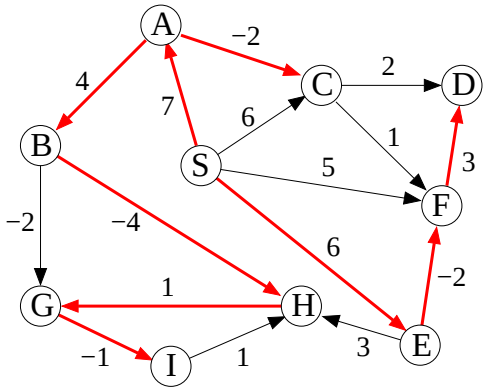
```
int double(int n)
 // input condition: n >= 0
 {
  int p = n;
  int q = 0;
  while(p > 0)
   {
    p = p-1;
    q = q+2;
   }
  return q;
 }
```

The function returns $2n$. The loop invariant is: $p \geq 0$ and $2p + q = 2n$.

2. Work Problem 4.2 on page 120 of your textbook.



	0	1	2	3	4	5	6
S	0	0	0	0	0	0	0
A	∞	7	7	7	7	7	7
B	∞	∞	11	11	11	11	11
C	∞	6	5	5	5	5	5
D	∞	∞	8	7	7	7	7
E	∞	6	6	6	6	6	6
F	∞	5	4	4	4	4	4
G	∞	∞	∞	9	8	8	8
H	∞	∞	9	7	7	7	7
I	∞	∞	∞	∞	8	7	7



The shortest path tree is shown in red.
There are two choices for the shortest path tree.

3. You can transform WARM to COOL in five steps by replacing one letter at each step, as shown below. This sequence is sometimes called a *ladder*. All words in the ladder must be common (not proper) English words. You would like to write a program which finds the shortest ladder between any two given English words, to be input by the user. You will make use of a list of all common 4-letter English words, which you find on the internet. Describe your program.

   Here are two solutions. The one I expected you to give is: Create an undirected graph G whose nodes are the 4-letter English words, and where there is an edge between two words if they differ by one letter. Solve the single source shortest path problem in G where WARM is the source, using breadth first search. The path to COOL as short as possible, and the backpointers will give you the ladder.

   You can take the same graph G, but use the A* algorithm where S = WARM and T = COOL, and where the $h(w)$ for any word w is the number of letters you need to replace to get COOL. For example, h(HERE) = 4 and h(BOLL) = 2.