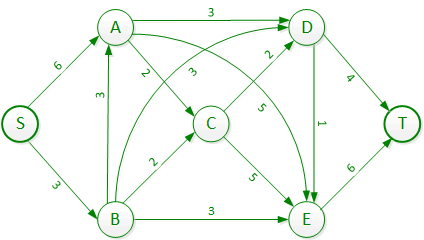
Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)

Homework B (*14/15* pts)

1. [*9/10* pts] Consider the following graph of a network with given flows:



Part A [4 pts]: Use the Edmonds-Karp algorithm to find the maximum flow from S to T in the graph below. On each line given in the table, list the path and the total augmented flow along that path. At the end, list the total flow through the network. When choosing among two different paths that are equally valid choices for the next augmentation in the Edmonds-Karp algorithm, choose the path that augments the flow by the greatest amount. (If you do this correctly, there is only one right answer). You may not need all four rows. List the total flow through the network.

|  |  |
| --- | --- |
| **Path** | **Augmented Flow** |
| SAET | 5/6-5/5-5/6 = 5 |
| SBDT | 3/3-3/3-3/4 = 3 |
| SADT | 6/6-1/3-4/4 = 1 |
|  |  |
| **TOTAL FLOW** | 5 + 3 + 1 = 9 |

Part B [1 pt]: In the table below, list the flow along each edge at the maximum network flow.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Edge | S🡪A | S🡪B | A🡪C | A🡪D | A🡪E | B🡪A | B🡪C | B🡪D | B🡪E | C🡪D | C🡪E | D🡪E | D🡪T | E🡪T |
| Flow | 6/6 | 3/3 | 0/2 | 1/3 | 5/5 | 0/3 | 0/2 | 3/3 | 0/3 | 0/2 | 0/5 | 0/1 | 4/4 | 5/6 |

Part C [2 pts]: In the table below, list the residual capacity of each edge in each direction in the final residual graph.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Edge | S🡪A | S🡪B | A🡪C | A🡪D | A->E | B🡪A | B🡪C | B🡪D | B🡪E | C🡪D | C🡪E | D🡪E | D🡪T | E🡪T |
| Flow | 0 | 0 | 2 | 2 | 0 | 3 | 2 | 0 | 3 | 2 | 5 | 1 | 0 | 1 |
| Edge | A🡪S | B🡪S | C🡪A | D🡪A | E🡪A | A🡪B | C🡪B | D🡪B | E🡪B | D🡪C | E🡪C | E🡪D | T🡪D | T🡪E |
| Flow | 6 | 3 | 0 | 1 | 5 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 4 | 5 |

Part D [1 pt] : In the final residual graph, which other nodes are reachable from source node S?

\_\_\_There are no node are reachable from S because the residual capacity of edges from source node S go to other nodes are all zero.\_

Part E [1 pt]: In the final residual graph, from which other nodes can you reach sink node T?

\_\_\_A B C D E\_\_because on edge E->T residual capacity still left 1 and other node still connect togetther and still have left some number capacity.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part F [1 pt]: Which edges in the original graph cross the maximum cut formed by the residual graph at the end of the Edmonds-Karp algorithm?

\_\_\_A->D\_A->C B->C B->E\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*S🡪A, S🡪B (-1)*

1. [5/5 pts] Consider the following problem. Using no more space than the next page, describe how you would solve it (describe an algorithm you would invent to do so, which may or may not involve using one or more of the algorithms we’ve used this year as subroutines).

Suppose that you have the problem of serving a large number of users with live streaming content from a small number of servers, in an environment in which the flow to a given consumer will come from exactly one server. At any given time you have some ongoing streams, and you know which consumers have requested new streams.

You want to assign paths through the network so that if possible, you can serve all requests. If this is not possible, you want to serve as many requests as possible.

Note a few things:

* Each user may request different content than each other user. So each stream is independent of each other stream.
* Any server can serve any steam at any requested rate.
* Once a server is selected to give content to a user, that user will continue to get content from that server – they can’t “switch servers in midstream”.
* Different requests may have a different bit rates, taken from a small selection of possibilities (some people may request higher definition than others, for instance).

Explain what algorithm(s) you would use or develop to solve this problem, and how you would do them.

Answer:

First step, to handle problem between users and servers connection, I will build a **Bipartite Matching as network flow graph** that have user’s nodes and server’s nodes, start node and sink node separate to both side, then each edges that connect between them have some value of requested rate. So between the two side, users and servers, we have to provide the most maximum of ability of each stream that connect to each server independent as each user may request follow by any requested rate. So I would like to use is **Edmonds-Karp algorithm** to find the maximum flow or maximum requested rate that following different requests from user. Because, this algorithm can provide the solution of priority taking a small selection of possibilities paths of a connection between user stream to server.