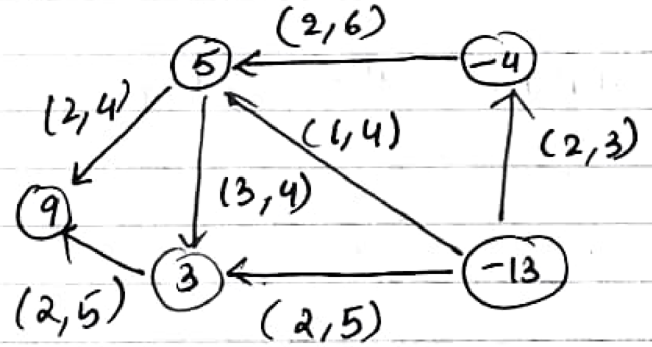
HomeWork 9

Ques. 1. In the network G below, the demand values are shown on vertices (supply value if negative). Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge. Determine if there is a feasible circulation in this graph. You need to show all your steps.



(a) Reduce the Feasible Circulation with Lower Bounds problem to a Feasible Circulation problem without lower bounds.

(b) Reduce the Feasible Circulation problem obtained in part a to a Maximum Flow problem in a Flow Network.

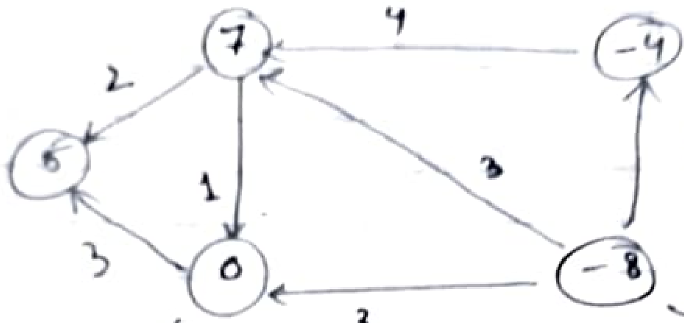
(c) Using the solution to the resulting Max Flow problem explain whether there is a Feasible Circulation in G.

C

Ans. (a)

5 - (3 - 5)
= 7

- 4 - (2 - 2)
= - 4

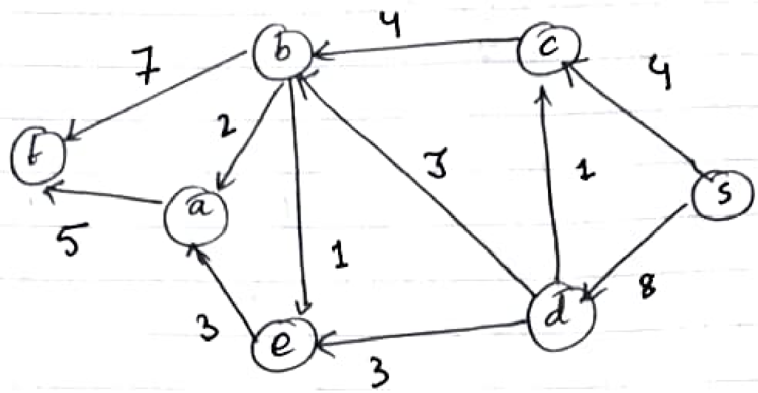


9 - (4)
= 5

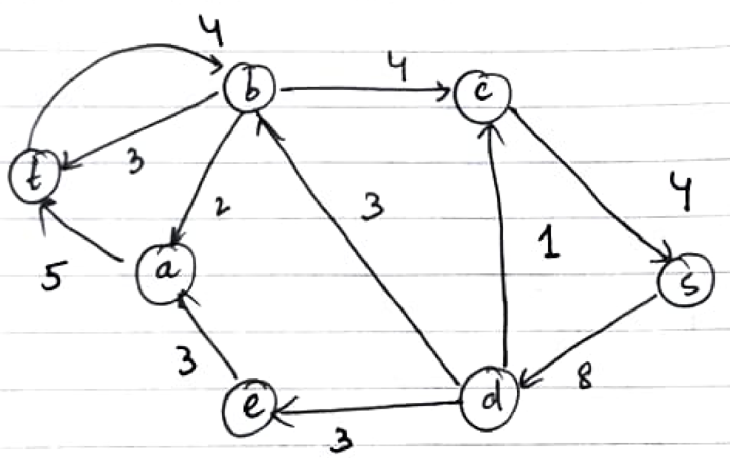
-13 - (0 - 5)
= -8

3 - (5 - 2)
= 0

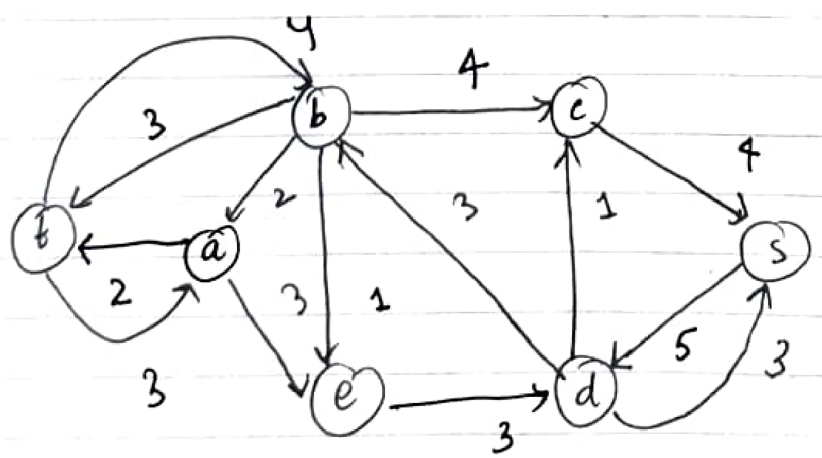
(b)



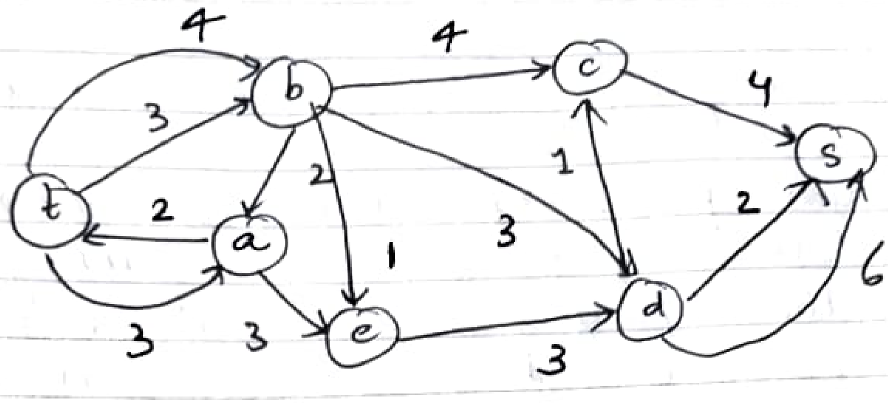
(c)



S → c → b → t , flow = 4



s → d → e → a → t , flow = 3



$s \rightarrow d \rightarrow b \rightarrow t$, flow = 3

Max-flow = 10

Since the value of Max Flow is less than the total demand value D = 12, there is NO Feasible solution in the circulation network, and therefore there is no feasible circulation in the circulation with lower bounds network.

Ques2. Solve Kleinberg and Tardos, Chapter 7, Exercise 31.

Some of your friends are interning at the small high-tech company Web-Exodus. A running joke among the employees there is that the back room has less space devoted to high-end servers than it does to empty boxes of computer equipment, piled up in case something needs to be shipped back to the supplier for maintainence.
A few days ago, a large shipment of computer monitors arrived, each in its own large box; and since there are many different kinds of monitors in the shipment, the boxes do not all have the same dimensions. A bunch of people spent all these things, realizing of course that less space would be taken up if some of the boxes could be nested inside others.
Suppose each box i is a rectangular parallelepiped with side lengths equal to (i_1, i_2, i_3); and suppose each side length is strictly between half a meter and one meter.
Geometrically, you know what it means for one box to nest inside another: It's possible if you can rotate the smaller so that it fits inside the larger in each dimension. Formally, we can say that box i with dimension (i_1, i_2, i_3) nests inside box j with dimensions $\{j_1, j_2, j_3\}$ if there is a permutation a, b, c of the dimensions $\{1, 2, 3\}$ so that $i_a < j_1$,

and $ib < j2$, and $ic < j3$. Of course, nesting is recursive: If i nests in j, and j nests in k, then by putting i inside j inside k, only box k is visible. We say that a nesting arrangement for a set of n boxes, is a sequence of operations in which a box i is put inside another box j in which it nests, and if there were already boxes ~~insie~~ nested inside i, then these end up inside j as well. (Also notice the following: Since the side lengths of i are more than half a meter each, and since side lengths of j are less than a meter each, box i will take up more than half of each dimension of j, and so after i is put inside j, nothing else can be put inside j.) We say that a box k is visible in a nested arrangement if the sequence of operations does not result in its ever being put inside another box.

Here is the problem faced by the people at Web Exodus: Since only the visible boxes are taking up any space, how should a nesting arrangement be chosen so as to minimize the no. of visible boxes?

Give a polynomial-time algorithm to solve this problem.

Example. Suppose there are 3 boxes with dimension (.6, .6, .6), (.75, .75, .75) and (.9, .7, .7). The first box can be put into either of the second or third boxes, but in any nesting arrangement, both the second and third boxes will be visible. So the minimum possible no. of visible boxes is two, and one solution ~~this is to put the first box inside~~

achieves this is to nest the first box inside the second.

Ans. We reduce the given problem to a circulation flow with lower bound problem where units of flow correspond to sets of boxes nested inside one visible box. We construct the following graph G:

- For each box i, G has two nodes u_i and v_i and an edge between them that corresponds to this box. This edge (u_i, v_i) has a lower bound of 1 and a capacity of 1. (Each box is exactly ~~or~~ in one set of boxes nested one in another.)
- For each pair of boxes, i and j, if box j can nest inside box i, there is an edge (v_i, u_j) with a lower bound of 0 and a capacity of 1 (One can store box j inside i).
- G also has a source node s, with demand $-k$ (corresponding to the back room where boxes are stored) and a sink node t, with demand k (corresponding to nothing inside empty boxes).
- For each box i, G has an edge (s, u_i) with a lower bound of 0 and a capacity of 1. (Any box can be visible).
- For each box j, G has an edge (v_j, t) with a lower bound of 0 and a capacity of 1 (Any box can be empty).

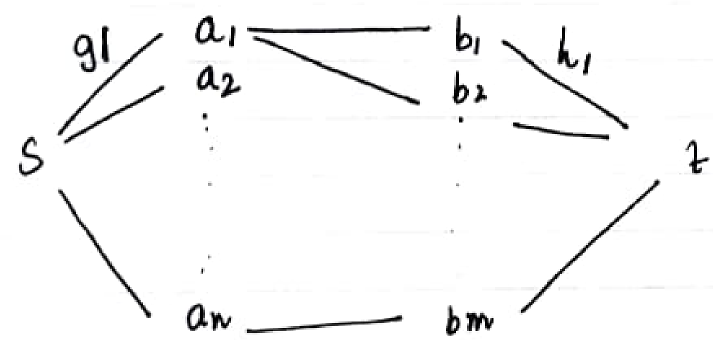
We claim the following: There is a nesting arrangement with k visible boxes if and only if there is a feasible circulation in G with demand $-k$ in the source node s and demand k in the sink t.

Proof :

- First, suppose there is a nesting arrangement with k visible boxes. Each sequence of nested boxes inside one visible box $i1, i2, \ldots, in$ defines a path from s to t: $(s, u_{i1}, v_{i1}, u_{i2}, v_{i2}, \ldots u_{in}, v_{in}, t)$ Therefore we have k paths from s to t.
- The circulation corresponding to all these paths satisfy all demands, capacity and lower bound. Conversely, consider a feasible circulation in our network. Without lost of generality, assume that this circulation has integer flow values.
- There are exactly k edges going to t that carry one unit of flow. Consider one of such edges (v_i, t). We know that (u_i, v_i) has one unit of flow. Therefore, there is a unique edge into u_i that carries one unit of flow. If this edge is of the kind (v_j, u_i) then put box i inside j and continue with box j. If this edge of the kind (s, u_i), then put the box i in the back room.
- This box became visible. Continuing in this way we pack all boxes into k visible ones. So we can answer the question whether there is a nesting arrangement with exactly k visible boxes.
- Now to find the minimum possible number of visible boxes we answer this question for $k=1,2,3$ and so on, until we find a positive answer.
- The maximum number of this iteration is n, therefore the algorithm is polynomial since we can find a feasible circulation in polynomia

Ques. 3. At a dinner party, there are n families $\{a_1, a_2, \ldots, a_n\}$ and m tables $\{b_1, b_2, \ldots, b_m\}$. The ith family a_i has g_i members and the jth table b_j has h_j seats. Everyone is interested in making new friends and the dinner party planner wants to seat people such that no two members of the same family are seated in the same table. Design an algorithm that decides if there exists a seating assignment such that everyone is seated and no two members of the same family are seated at the same table.

Ans.



- a_i - vertex corresponding to ith family
- b_j - vertex corresponding to jth table
- From every family vertex a_i to table vertex b_j, add edge (a_i, b_j) of capacity 1.
- Add vertex s and t.
- To every family vertex a_i, add an edge (s, a_i) of capacity g_i
- From every table vertex b_j add edge (b_j, t) of capacity h_j.

- There exists a valid seating if value of max from from s to t in the above network equals $g_1 + g_2 + \ldots g_n$.

Proof of claim:

- Assume there exists a valid seating that is a seating where every one is seated and no two members in a family are seated at a table.
- We construct a flow f do the network as follows -
- If a member of the ith family is seated at the jth table in the seating assignment, then assign a flow of 1 to the edge (a_i, b_j).
- Else assign a flow of 0 to the edge (a_i, b_j).
- The edge (s, a_i) is assigned a flow equaling the number of members in the ith family that are seated (which since the seating is valid equals g_i).
- Likewise the edge (b_j, t) is assigned a flow equaling the number of seats taken in the table b_j (which since the seating is valid ~~since the~~ by is at most h_j.)
- Clearly the assignment is valid since by construction the capacity and conservation constraints are satisfied.
- Further, the value of the flow equals $g_1+g_2+\dots+g_n$.

→

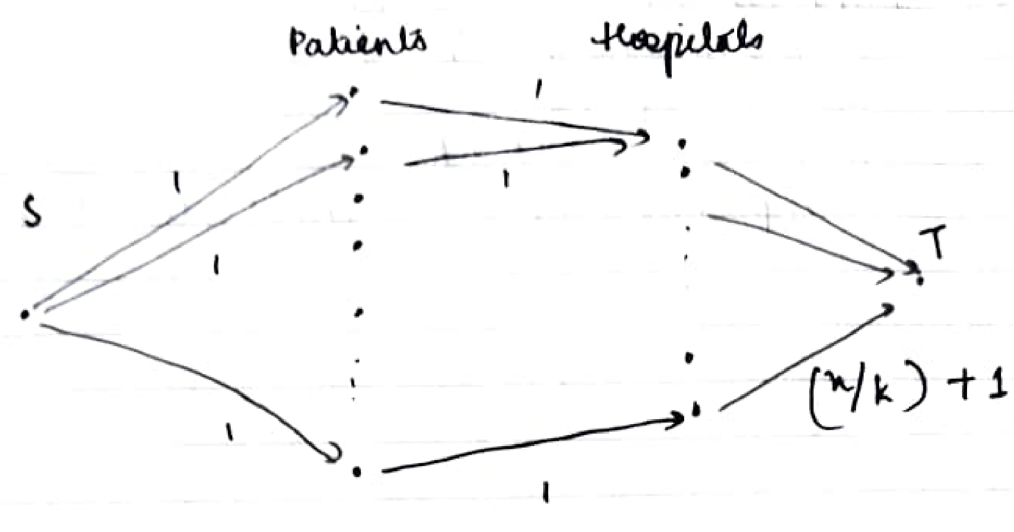
- Conversely, assume that the value of the max s-t flow equals $g_1+g_2+\cdots+g_n$.
- Since the capacities are integers, by the correctness of the Ford-Fulkerson algorithm, there exists a maxflow (call f) such that the flow assigned to every edge is an integer.
- In particular, every edge between the the family vertices and table vertices has a flow of either 0 or 1 (since these edges are of capacity 1).
- Seating arrangement - Seat a person of the ith family at the jth table if and only if $f(a_i, b_j)$ is 1. By construction at most one member of a family is seated at a table.
- Since the value of f equals the capacity of the cut $(\{s\}, V-\{s\})$, every edge out of s is saturated.
- Thus, by flow conservation of a_i, for every a_i the number of edges out of a_i with a flow of 1 is g_i.
- Thus in the seating assignment, every one is seated. Further, since the flow $f(b_j, t)$ out of b_j is at most h_j, at most h_j persons are seated at table b_j. Thus we have a valid seating.

Ques. 4. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour drive to their current location. (So different patients will be able to be served by different hospitals depending upon the patient's locations.) However, overloading one hospital with too many patients at the same time is undesirable, so we would like to distribute the patients as evenly as possible across all the hospitals. So the paramedics (or a centralized service advising the paramedics) would like to work out whether they can choose a hospital for each of the injured people in such a way that each hospital receives at most $(n/k+1)$ patients.

(a) Describe a procedure that takes the given information about the patients locations (hence specifying which hospital each patient could go to) and determines whether a balanced allocation of patients is possible (i.e. each hospital receives at most $(n/k+1)$ patients).

(b) Provide proof of correctness for your procedure.

(c) What is the asymptotic running time of your procedure (in terms of n and k)?



connect patient to hospital with directed edge of capacity 1 if hospital is within the patient half-hour's drive.

(b) · Each unit flow from S to T is equivalent to assigning a patient to a hospital with Restriction

- each hospital get less than (n/k)+1 patients. Each patient will be assign to only one hospital which is located in the half-hour's drive to the patient.
- Ford-Fulkerson algorithm can be used to find maximum flow. The maximum flow is the maximum of this assignment and if we can assign all patients (max-flow = n) we can do balance allocation.

(c) Running time is $O(Cm)$

C - maximum possible flow

m - no. of edges.

$C = n$, $m = n + nk + k$

complexity - $O(n(n+k+nk)) = O(n^2k)$