Lab14 Solution&Hint

YAO ZHAO

Lab 13 Q2 Solution

opt[i][j][k] means i the first i number, j means the i-th number is j
if k = 1 means Ai<=Ai-1 Ai+1 has no constraint
if k = 0 means Ai>Ai-1 so Ai+1 must >= Ai

If Ai = -1 $opt[i][j][0] = \sum_{n=1}^{j-1}(opt[i-1][n][0] + opt[i-1][n][1])$ k=0 means Ai>Ai-1, for each j, j > [1,j-1]
$opt[i][j][1] = \sum_{n=j}^{200}(opt[i-1][n][1]) + opt[i-1][j][0]$ k=1, means Ai<=Ai-1, for each j, j<=[j,200]

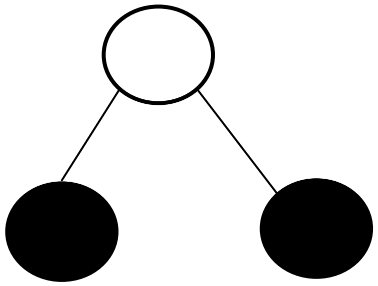
If Ai != -1 if Ai = m, $opt[i][m][0] = \sum_{n=1}^{m-1}(opt[i-1][n][0] + opt[i-1][n][1])$
$opt[i][m][1] = \sum_{n=m}^{200}(opt[i-1][n][1]) + opt[i-1][m][0]$
if Ai ≠ m, $opt[i][j][0] = 0$ $\text{opt}[i][j][1] = 0$
initial state: if A1 = -1 $opt[1][j][0] = 1, opt[1][j][1] = 0 (1 \le j \le 200)$ because $A1 \le A2$
if A1=m $opt[1][m][0] = 1,$ otherwise 0

terminal state: answer = $\sum_{m=1}^{200}(opt[n][m][1])$ *because* An<=An-1

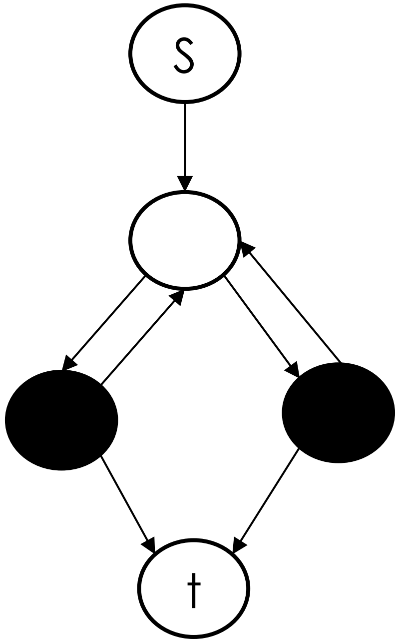
Lab14 Q1 Solution

- A classic network flow question
- Construct a new graph G’
  - First, create a source and a sink.
  - The points of 1 connect to the source point, the point of 0 connect the sink point (or reverse), and the edges in the original graph are bidirectional edges in the new graph.
  - All weight value of edges in new graph are 1
- The answer of original question is the min cut in G’

3 2
1 0 0
1 2
1 3



G

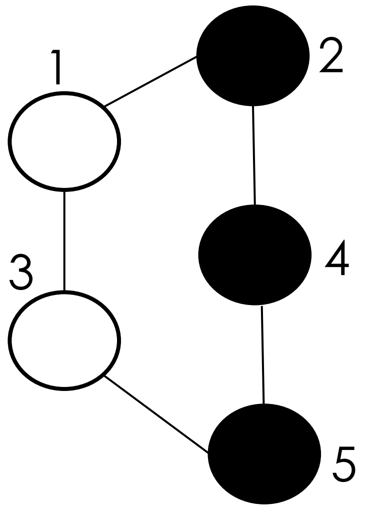


G’

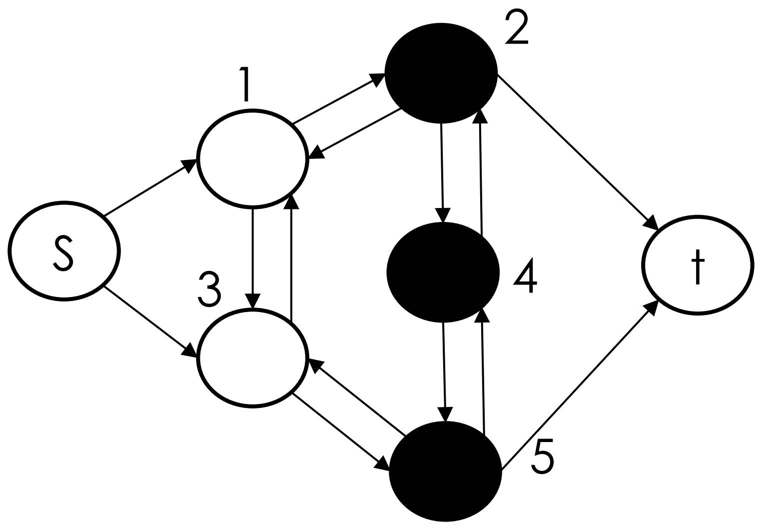
Lab14 Q1 Solution

A more complex sample：

```
5 5
1 0 1 0 0
1 2
3 1
4 2
4 5
3 5
```



G



G’

Lab 14 Q2 Solution

- First, figure out the number of classrooms: Greedy. Assume the number is m.
- Take a ball as an objects. m objects should be selected from n objects. In order to meet the requirements of the question, the sum of B[i] of these m boxes should be >= the total number of balls, and the number of balls already in these m boxes should be as large as possible (in order to minimize the number of balls need to move).

m objects should be selected from n objects. In order to meet the requirements of the question, the sum of B[i] of these m boxes should be >= the total number of balls, and the number of balls already in these m boxes should be as large as possible (in order to minimize the number of balls need to move).

VS

Two-dimensional backpack model:
There are N objects and a backpack with a capacity of W and a volume of D. The weight of the i-th item is w[i], the volume is d[i], and the value is v[i]. You should find which objects can be loaded into the backpack so that the total weight of these items does not exceed the backpack's capacity, the total volume does not exceed the backpack’s volume, and the total value is the largest.

In **Two-dimensional backpack model**, the sum of the two dimensions are always less than or equal to a constant, while this question requires the sum of the selected boxes capacity to be greater than or equal to a constant.

- Observe $1 \le A[i] \le B[i] \le 100$. Assuming that the total number of students is represented by S, and let a constant C >=100, the difference between B[i] and target C of each box can be represented as C-B[i].
- The original requirement: $\sum_{\mathrm{i}=1}^{m} B[i]$ >=S
- Convent to: $\sum_{\mathrm{i}=1}^{m}(C - B[i]) \le \mathrm{mC} - \mathrm{S}$
- Then the original question is converted a classic **Two-dimensional backpack problem.**