# Algorithm Design and Analysis: Homework 5

Due on Dec 19, 2017 at 10:15am

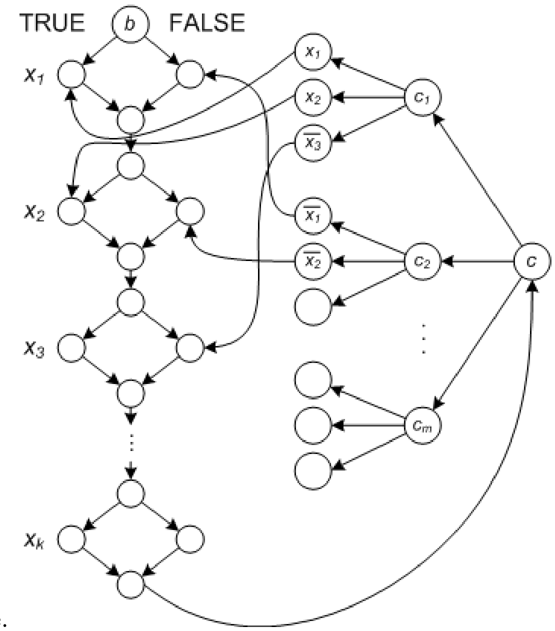
*14784547 luochenqi*

**Luochenqi**

# Problem 1

**Solutions**

It is in $PSPACE$,because the recursion stack is in polynomial because there are at most n levels since n is the number of all nodes.
the FORMULA-GAME problem consists of a quantified Boolean formula .




We can reduce Formula-game problem to GG in polynomial time.

# Problem 2

**Solutions**
Let $GG = \{< G, b > |P_1$has a winning strategy for the generalized geography game played on graph G starting at node b$\}$
Proof: 1.Measure the out-degree of node $n_{start}$.If this degree is 0,then return reject,because there are no available path for player to move.
2.construct a list of all nodes reachable from $n_{start}$ by one edge,$n_1, n_2..., n_i$
3.remove $n_{start}$ and all edges connected to it from G to form $G_1$.
4.for each node $n_i$ in the list,call $M(< G_1, n_i >)$.
5.if all these calls return accept ,then no matter which decision $P_1$ makes,$P_2$ has a strategy to win,so return reject.Otherwise if one of the call returns reject,$P_1$ got a choice to deny any successful strategies for $P_2$,so return accept.

# Problem 3

**Solutions**
We can start at the leaves of the tree.Since any nodes $\in V - S$ have an edge to the dominating set.So the dominating set includes either leaves' parent or all the leaves.So it is a dynamic programming problem.
For all nodes u of G in post-order,if u is a leaf then set the value
$M_{out}[u] = 0$

Problem 3 continued on next page. . . 

$M_{in}[u] = cost_u$
else $M_{out}[u] = \sum_{v\in children(u)} min(M_{out}[u]M_{in}[u])$
$M_{in}[u] = cost_u + \sum_{v\in children(u)} M_{out}[u]$.
And finally we return $min(M_{out}[root], M_{in}[root])$

## Problem 4

**Solutions**
consider each edge as a triangle which we match,similar to a bipartite matchingit starts with adding triangles until we cannot add.
maximal 3d-matching is a 3-approximation because each edge in the maximal matching could have replaced three edges in the maximum matching . step1:consider any triple in the matching set.
step2:use matching set and all its neighbor triples.
step3:while T!=empty T=M;$M = M \cup \{(x, y, z)\}$T=T-M

## Problem 5

**Solutions**
we use first-fit algorithm ,

```
b=0
for i=1 to n do
  let j be first bin that can fit object i with size Si*
  if j exist then
   insert i to L[i]
   B[j]=b[j]-Si
  else
    b=b+1
    insert i to list[b]
    B[b]=1-Si
```

$b = \frac{total\ space\ in\ bins}{size\ of\ each\ bin} = \frac{space\ used\ in\ bins+space\ unused\ in\ bins}{1} = S + e \geq S$,since b is an integer.if we assume b=1,b=s+e$\geq S + \frac{b}{2}$,so $b \leq 2S \leq [2S]$.Let $b^*$ be the optimal number of bins.$[S] \leq b^*$,$b \leq [2S] \leq 2[S] \leq 2b^*$,so $\frac{b}{b^*} \leq 2$,so the approximation ratio is 2.

## Problem 6

**Solutions**

we set an indicator $X_e = 1_{satisfied}$,for e∈E,then we compute using linearity of expectation. E[number of satisfied edges]=$E[\sum_{e\in E}]$=$\sum_{e\in E} E[X_e]$=$\sum_{e\in E} P[satisfied] = 2/3[E]$
the tpimal number of satisfied edges canbe no more than total number of edges,$c^* \leq |E|$ ,$\frac{2}{3}|E| \geq \frac{2}{3}c^*$