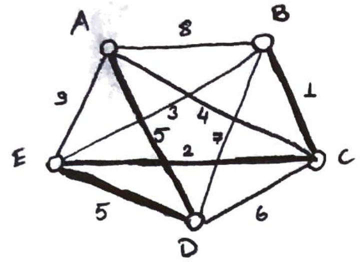
Edin Guso
23435

CS407 HW7

6.2.3.:



Solution: B-C-E-D-A

Cost: 1+2+5+5=13

6.2.4.:

Consider the Clique problem to prove this,

(⇒) If there is a polynomial time algorithm for the original problem, there is one for the yes-no problem:

Assumption is that we can find a solution to the problem in polynomial time. Then, we just need to find the size of the solution in polynomial time. Now, we can answer the yes-no problem.

(⇐) If there is a polynomial time algorithm for the yes-no problem, there is one for the original problem:

Assumption is that we can answer the yes-no problem in polynomial time. We can find the largest clique with size k by gradually increasing k. Then, we know that there is a clique with size k. Not all of the vertices in V are in the clique. Call the subset of vertices that are part of the clique V_c. Our goal is to find this V_c. Assume our initial graph is $G=(V,E)$ and $e(v)$ denotes the edges adjacent to vertex $v \in V$. Apply the following algorithm:

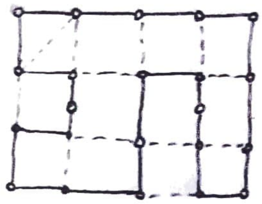
```
Vc = {}
while (V ≠ Vc) {   //until all remaining vertices are used in the clique
    Select v ∈ V-Vc   //select a random vertex
    V' = V-{v}
    E' = E-e(v)
    if( G'=(V',E') has a clique of size k) {   //does the remaining graph have a clique
        V = V'
        E = E'   //if so, this vertex is not used in the clique
    }
    else {
        Vc = Vc ∪ {v}   //otherwise, this vertex is used in the clique.
    }
}
return (V,E).
```

This is a polynomial time algorithm that finds the solution if the yes-no version is polynomial time.

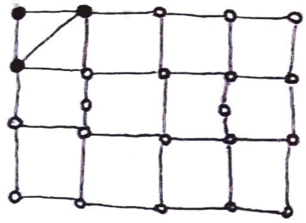
Edin Guso
23435

6.2.5 :

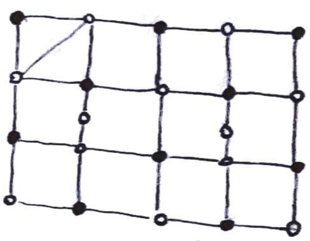
It does have a hamiltonian cycle:



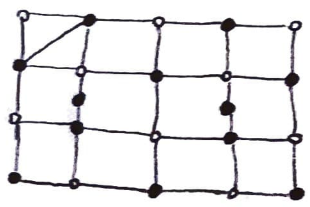
Largest clique has size 3:



Largest independent set has size 10:



Smallest node cover has size 12:



6.3.1:

x_1	x_2	x_3	x_4	is Solution?
$\bot$	$\bot$	$\bot$	$\bot$	Yes
$\bot$	$\bot$	$\bot$	$\top$	Yes
$\bot$	$\bot$	$\top$	$\bot$	Yes
$\bot$	$\bot$	$\top$	$\top$	No
$\bot$	$\top$	$\bot$	$\bot$	No
$\bot$	$\top$	$\bot$	$\top$	No
$\bot$	$\top$	$\top$	$\bot$	Yes
$\bot$	$\top$	$\top$	$\top$	Yes
$\top$	$\bot$	$\bot$	$\bot$	No
$\top$	$\bot$	$\bot$	$\top$	Yes
$\top$	$\bot$	$\top$	$\bot$	No
$\top$	$\bot$	$\top$	$\top$	No
$\top$	$\top$	$\bot$	$\bot$	No
$\top$	$\top$	$\bot$	$\top$	Yes
$\top$	$\top$	$\top$	$\bot$	No
$\top$	$\top$	$\top$	$\top$	Yes

There are 8 solutions to the given problem.

Edin Guso
23435

6.3.2:

a) If there is a clause with a single literal, we must purge this literal. We always purge when we have this condition (set the single literal to true, remove all clauses that include this literal and remove the complement of this literal from each 2-literal clause that contains it).

If there isn't a literal that can be purged, pick a random variable and assign it to T, then start purging. If it fails try $\perp$. If both fail, no solution. If either succeed that is the solution. It is clear that this algorithm correctly solves 2-SAT. It is also obvious that it is in P since every literal will be assigned at most 2 values and each purge is polynomial time.

b) If n is the number of literals and c is the number of clauses, the lower bound is:

$$O(2 \cdot n \cdot c) = O(nc)$$

c) i) Try $T(x_1) = T$

$\rightarrow (\bar{x}_4), (x_2 \vee \bar{x}_3), (x_3 \vee x_4)$

ii) Set $T(\bar{x}_4) = \perp$

$\rightarrow (x_2 \vee \bar{x}_3), (x_3)$

iii) Set $T(x_3) = T$

$\rightarrow (x_2)$

iv) Set $T(x_2) = T$

Solution: $T(x_1) = T$, $T(x_2) = T$, $T(x_3) = T$, $T(x_4) = \perp$

6.3.3:

We know that the clause (x) can be thought of as $(\bar{x} \rightarrow x)$.

Also, we know that $x \rightarrow y$ and $y \rightarrow z$ imply $x \rightarrow z$. Therefore, having a path from x to z is equivalent to having the condition $x \rightarrow z$ itself.

$\Leftarrow$: There is a path from x to $\bar{x}$ and from $\bar{x}$ to x.

We know that this is equivalent to having $x \rightarrow \bar{x}$ and $\bar{x} \rightarrow x$ condition which is equivalent to having clauses (x) and $(\bar{x})$ which is unsatisfiable.

$\Rightarrow$: 2-SAT is unsatisfiable.

Assume that there isn't a path from x to $\bar{x}$ and $\bar{x}$ to x. Then, we cannot create a logical contradiction, meaning that the problem is satisfiable. Contradiction!