Intro to Analysis of Algorithms CSE 4081 VERSION UG

Fall 2015 Exam 3 Points 40 Time 70 min

*[Students ignore these key words: Foundations, Problem solving, Algorithm understanding.]*

**Q1a.** LP

Minimize (-6x1 -2x2 -4x3)

Such that: 2x1 +2x2 +6x3 ≤ 60

-4x1 -4x2 -10x3 ≥ -48

8x1 +2x2 +4x3 ≤ 72

where: x1, x2, x3 ≥ 0 [7+1+2]

*Standard Form*

*Max: 6x1 + 2x2 + 4x3*

*Constraints:*

*2x1 + 2x2 + 6x3 <= 60*

*4x1 + 4x2 + 10x3 <= 48*

*8x1 + 2x2 + 4x3 <= 72*

*Where x1, x2, x3 >= 0*

*Equivalent Slack Form:*

*z = 0 + 6x1 + 2x2 + 4x3*

*Linear Equations:*

*x4 = 60 – (2x1 + 2x2 + 6x3)*

*x5 = 48 – (4x1 + 4x2 + 10x3)*

*x6 = 72 – (8x1 + 2x2 + 4x3)*

*Constraints:*

*xi >= 0, i = 1, 2, 3, 4, 5, 6.*

*Sets: N = {x1, x2, x3}, B = {x4, x5, x6}*

*Entering variables xe in pivot, let’s say xe = x1. So x6 is the leaving variable (the most constraining basic variable).*

*Rewrite:*

*To:*

*Then: z = 0 + 6x1 + 2x2 + 4x3 will be*

*And:*

*Then x2 is chosen to be the entering variable, then the leaving variable is x5.*

*Similarly,*

*So,*

**Q1b.** Is LP in NP-class?

*Yes, it is in P-class and hence in NP-class. Polynomial algorithms do exist (Kahachien’s, Karmarkar’s, and other interior point algorithms).*

**Q1c.** Write a dual LP for the above problem.

*Check LP slide number 33.*

**Q2a.** Create the output 3SAT problem for the following SAT problem instance. [4+6]

|  |  |
| --- | --- |
| **Input SAT-clause** | **Output 3SAT clause** |
| (~a, b) *T* | (~a, b, z1), (~a, b, ~z1) *T: z1=do-not-care T/F* |
| (a, c, d) *T* | (a, c, d) *T* |
| (g) *F* | (g, z2, z3), (g, z2, ~z3), (g, ~z2, z3), (g, ~z2, ~z3)  *z2=T, z3=T: show that clauses=T, T, T, F*  *z2=T, z3=F: show that clauses=T, T, F, T*  *z2=F, z3=T: show that clauses=T, F, T, T*  *z2=F, z3=F: show that clauses=F, T, T, T*  *All 4 3-clauses cannot be made T.* |
| (a, c, b, e, f, ~g)  *T* | (a, c, z4), (~z4, b, z5), (~z5, e, z6), (~z6, f, ~g)    *T: z4, z5, z6: do-not-care* |
| (b, ~c, d, e, ~f)  *T* | (b, ~c, z9), (~z9, d, z10), (~z10, e, ~f)  *T: z9, z10 = do-not-care* |

**Q2b.** Below is a set of variable assignments for the above source SAT problem:

(a=T, b=T, c=T, e=T, d=F, ~~e=F~~, f=F, g=F).

Try to use the new 3SAT variables’ assignments to show that the *T/F* remains the same for each row. That is, if the source-side clause is *T* with the above assignments, then all corresponding target side clauses in the same row can be made *T*; and if the source-side clause is *F* with the above assignments, then all corresponding target side clauses cannot be made *T* with respective new variables’ assignments.

*Use above values for old variables, and T+F values for each of new variables in each row. Show that when left column (source-SAT) is T, so is right column (target-3SAT), but when left column is F, then each assignments on right column is F.*

**Q3a.** Is Strongly-connected-component finding problem (SCC) an NP-hard problem? Briefly provide rationale to your answer. [2+8]

*No, the algorithm we have is polynomial. Complexity is not on any of my slides, but DFT-spanning tree creation is polynomial, I mentioned it many a times.*

**Q3b.** Find SCC in the following directed graph. Show results of intermediate steps, start with node *v1*.

**v1**

**v2**

**v3**

**v4**

**v6**

**v5**

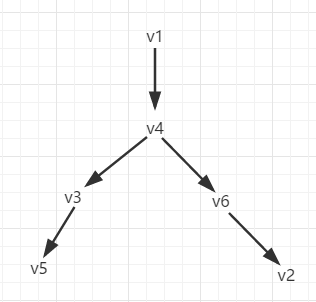
From v1 do DFS, then we have: v1(6)-v4(5)-v6(4)-v2(3)-v3(2)-v5(1)

*Provide the trees.*

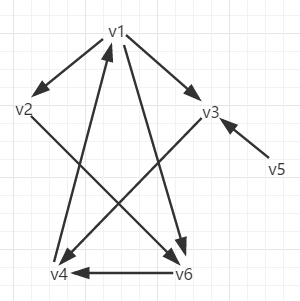
*Note, there could be many trees, even when started with v1.*

*No points be deducted if one starts with different node.*

*Tree may like this:*



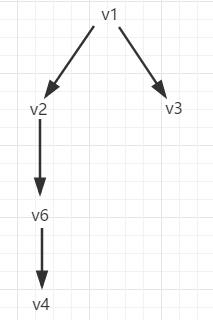
Create Gr



DFS on Gr, start with node v1 since v1 has the highest number. Then we have:

v1-v2-v6-v4-v3, and v5

v1 tree may like this:



**Q4a.** What action will you take if the graph below is a communication network, and you want to avoid the network getting disconnected when any node dies. For any algorithm you run, you must show the intermediate trees and graphs, start your algorithm with node *v1*. (Note that the action asked of you above is subjective, based on the result of your algorithm.) [8+2]

*Key:*

*Articulation Points Finding Alg:*

*Only one AP (v3).*

*Must show the DFT spanning tree with the (num, low) numbers and tree. Justify, why v3 is found to be AP.*

*Action: by-pass v3 by adding an arc in such a way that there is no AP in the graph, e.g. add (v5, v6).*

**Q4b.** Explain briefly how many steps will a Depth-first traversal (DFT) spanning tree-drawing algorithm go through on the graph below.

**v1**

**v2**

**v3**

**v4**

**v6**

**v5**

*Key: Could be n, or m, or n+m. I prefer you provide a number for the steps.*