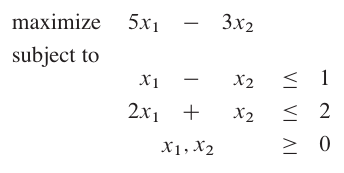
**Advanced Algorithms**

**Exercise for Lecture 13-15**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **周思源** | **Student ID** | 24S151092 |
| **Lecture 13** |  | | |
| **Lecture 14** |  | | |
| **Lecture 15** |  | | |
| **Total Score** |  | | |
| **Notes** | Deadline: **2024-10-31 24:00**  Submission Format: ‘**Lecture13-15\_Name\_Student ID.docx**’, and please send to: **aa\_24fall\_hw@163.com**  This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. | | |

**Lecture13**

**Problem 1. [50 points]**Solve the following linear program using SIMPLEX.



\z = 5x1 – 3x2

\x3 = 1 – x1 + x2

\x4 = 2 – 2x1 – x2

\x1,x2,x3,x4>=0

Get the basic solution (0, 0, 1, 2), z = 0;

Let xe = x1,

\z = 5 – 5x3 + 2x2

\x1 = 1 – x3 + x2

\x4 = 2x3 - 3x2

\x1,x2,x3,x4 >= 0

Get the basic solution (1, 0, 0, 0), z = 5;

Let xe = x2,

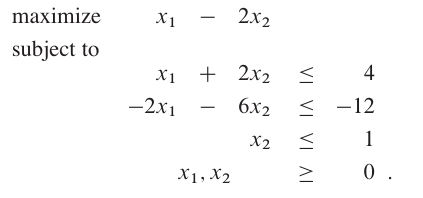
\z = 5 – 11/3x3 – 2/3x4

\x1 = 1 – 1/3x3 + 1/3x4

\x2 = 2/3x3 – 1/3x4

Get the basic solution (1, 0, 0, 0), z = 5

**Problem 2. [50 points]**Solve the following linear program using SIMPLEX.



Can not find the basic solution, form a auxiliary linear program,

Max -x0

Subject to :

\x1 + 2x2 – x0 <= 4

\-2x1 – 6x2 – x0 <= -12

\x2 - x0<= 1

\x1,x2,x0>=0

Slack form:

\z = -x0

\x3 = 4 – x1 – 2x2 + x0

\x4 = -12 + 2x1 + 6x2 + x0

\x5 = 1 – x2 + x0

\x0,x1,x2,x3,x4,x5>=0

Let xe = x0,

\z= -12 + 2x1 + 6x2 – x4

\x3 = 16 – 3x1 – 8x2 + x4

\x0 = 12 – 2x1 – 6x2 + x4

\x5 = 13 – 2x1 – 7x2 + x4

\x0,x1,x2,x3,x4,x5>=0

(12,0,0,16,0,13), z = -12,xe = x1

\z = -4/3 + 2/3x2 – 2/3x3 – 1/3x4

\x1 = 16/3 – 8/3x2 – 1/3x3 + 1/3x4

\x0 = 4/3 – 2/3x2 + 2/3x3 + 1/3x4

\x5 = 7/3 – 5/3x2 + 2/3x3 + 1/3x4

\x0,x1,x2,x3,x4,x5>=0

Let xe = x2

\z = -2/5 – 2/5x3 – 1/15x4 – 2/5x5

\x1 = 8/5 – 7/5x3 – 1/5x4 + 8/5x5

\x0 = 2/5 + 2/5x3 + 1/15x4 + 2/5x5

\x2 = 7/5 + 2/5x3 + 1/5x4 – 3/5x5

Find the basic solution (2/5, 8/5, 7/5, 0, 0, 0), z = -2/5,the linear program is unfeasible.

**Lecture14**

**Problem 1. [40 points]** What is the FFT of (1, 0, 0, 0) What is the appropriate value of ω in this case? And of which sequence is (1, 0, 0, 0) the FFT?

**Solution:**

To compute the FFT of the sequence (1, 0, 0, 0), we can start by:

ω=e^(−2πi/N)

We can get the N = 4, so ω=e^(−2πi/4)=e^(−πi/2)=−i

The even part (1, 0), the odd part (0, 0)

FFT of (1, 0) = (1, 1), FFT of (0, 0) = (0, 0)

X[0] = 1 + 0 = 1, X[1] = (1 + 0) + 0\*(-i) = 1, X[2] = (1 + 0) + 0\*(-1) = 1, X[3] = (1 + 0) + 0\*I = 1

The FFT of (1, 0, 0, 0) is (1, 1, 1, 1)

The sequence (1, 1, 1, 1) is actually the Inverse FFT (IFFT) of (1, 0, 0, 0).

**Problem 2. [60 points]**

**1.** Suppose that you want to multiply the two polynomials x + 1 and x2 + 1 using the FFT. Choose an appropriate power of two, find the FFT of the two sequences, multiply the results componentwise, and compute the inverse FFT to get the final result.

**2.** Repeat for the pair of polynomials 1 + x + 2x2 and 2 + 3x.

**Solution:**

**1.F(x) = x + 1, G(x) = x^2 + 1**

**The degree of F(x) is 1, G(x) is 2**

**So 1 + 2 = 3, N = 4;**

**F(x) (1, 1, 0, 0), G(x)(1, 0, 1, 0)**

**FFT(1, 1, 0, 0):  
X[0] = 1 + 1 + 0 + 0 = 2, X[1] = 1 + 1\*(-I) + 0\*(-1) + 0\*(-1) = 1 – I, X[2] = 1 + 1\*(-1) + 0 \* (-1) + 0\*(-1) = 0, X[3] = 1 + 1 \* I + 0 \* I + 0 \* I = 1 + i**

**FFT(1, 0 , 1, 0):**

**X[0] = 1 \* 1 + 0 \* 0 + 1 \* 1 + 0 \* 1 = 2, X[1] = 1 + 0 \* (-i) + 1 \* (-1) + 0 \* I = 0, X[2] = 1 \* 1 + 0 \*(-1) + 1 \* 1 + 0 \* (-1) = 2, X[3] = 1 \* 1 + 0 \* (-i) + 1 \* (-1) + 0 \* (-i) = 0**

**Z[i] = Xf[i] \* Xg[i]**

**So Z = (4, 0, 0, 0)**

**FFT^(-1) \*(4, 0, 0, 0)^T = (1, 1, 1, 1)^T**

**So is 1 + x + x^2 + x^3**

**2.F(x) = 1 + x + 2x^2, G(x) = 2 + 3x**

**F(x) : (1, 1, 2, 0) G(x) : (2, 3, 0, 0)**

**FFT(1, 1, 2, 0):**

**X[0] = 1 + 1 + 2 + 0 = 4, X[1] = 1 + 1 \* I + 2 \* (-1) + 0 \* (-i) = -1 + I, X[2] = 1 + 1 \* (-1) + 2 \* (1) + 0 \* (-i) = 2, X[3] = 1 + 1 \* (-i) + 2 \* (-1) + 0 \* (i) = -1 – i**

**FFT(2, 3, 0, 0):**

**X[0] = 2 + 3 + 0 + 0 = 5, X[1] = 2 \* 1 + 3 \* I + 0 + 0 = 2 + 3i, X[2] = 2 \* 1 + 3 \* (-1) + 0 + 0 = -1, X[3] = 2 \* 1 + 3 \* (-i) + 0 + 0 = 2 – 3i**

**Z = (20, -5 – I, -2, I - 5)**

**FFT^(-1) \* (20, -5 – I, -2, I - 5) = (2, 5, 7, 6)**

**So the poly is 2 + 5x + 7x^2 + 6x^3**

**Lecture15**

**Problem 1.[60 points]**

Please prove that clique problem is NPC. Given that 3-SAT problem is a NPC problem.

For a given undirected graph G=(V,E), the clique of it is V’ which is a subset of V. For any point pair u,v∈V’, the edge (u,v)∈E. And the clique problem is to find the max clique which contains most points of the graph G.

**Solution:**

Given an instance and a certificate , the validation requires checking each pair of vertices in V to see if there is an edge between them. This requires O(n^2) time, so clique is NP.

Given an instance C of 3CNF formula and a clique certification of size k, construct a graph G using positive integer k such that G has a clique of size of k if C is satisfiable.

For each clause of C, add a group of 2 vertices to G and add an edge between two vertices if they are in different triples and if the literals are consistent. Choose k equals to the number of clauses in C. the graph G can be constructed in time polynomial in the number of clauses in C.

Suppose y is a satisfying truth assignment for C. then each clause in C has at least one literal that is assigned true when y is applied and each such literal corresponds to a vertex in a triple. Choosing one triple of connected vertices in G represents a true literal from each clause and yields a set V of k vertices. To see that the set V is a clique , note that each node in G representing a true literal is connected to the nodes representing true literals in other clauses because it cannot be the connected nodes are complements. So the transformation yields, a k-clique on G.

Suppose f(y) is a clique V of size k on G. no edges in G connect vertices in the same triple and so V contains exactly one vertex per triple. We can create a truth assignment y by assigning 1 to each literal in C that corresponds to a vertex in V without the possibility of connecting a literal and its complement, since G contains no edges between inconsistent literals. Therefore, the clause of C satisfies under y and C belongs the 3-SAT.

**Problem2. [40 points]**

Define the optimization problem LONGEST-PATH-LENGTH as the relation that associates each instance of an undirected graph and two vertices with the number of edges in a longest simple path between the two vertices. Define the decision problem LONGEST-PATH ={⟨*G*,*u*,*v*,*k*⟩:*G*=(*V*,*E*) is an undirected graph, *u*,*v*∈*V*,*k*≥0 is an integer, and there exists a simple path from *u* to *v* in *G* consisting of at least *k* edges}. Show that the optimization problem LONGEST-PATH-LENGTH can be solved in polynomial time if and only if  LONGEST-PATH ∈ *P*.

**Solution:**

For the LONGEST-PATH is a simple path ,so k is bigger than 0 and smaller than |E|, we can solute the k by binary-search according the decision problem. So the decision problem algorithm time is O(lg|E|\*T1), so the LONGEST-PATH-LENGTH can be solve by go each pair of the graph in the set, so the algorithm can be solve in O(|V|^2lg|E|), so the two questions is can be reducation, so the optimization problem LONGEST-PATH-LENGTH can be solved in polynomial time if and only if  LONGEST-PATH ∈ *P.*