Northeastern University, Algorithms CS 5800 Fall 2016
Problem Set 6

Solutions to Problem Set 6

Rashmi Dwaraka

November 29, 2016

Contents

1 Alternatives to Strassens algorithm

Problem statement:
V.Pan has discovered a way of multiplying 68×68 matrices using 132,464 mul- tiplications, a way of multiplying 70 × 70 matrices using 143,640 multiplications, and a way of multiplying 72 × 72 matrices using 155,424 multiplications. Which method yields the best asymptotic running time when used in a divide-and-conquer matrix-multiplication algorithm? How does it compare to Strassens algorithm?

Solution:

The recurrence for divide-and-conquer matrix-multiplication algorithm can be written as below. Solving it using Master's Theorem, we can say:

(a) 68 × 68:

$$\begin{aligned} T(n) &= 132464 \cdot t(\frac{n}{68}) + n^2 \\ &= \Theta(n^{\log_{68} 132464}) \\ &\approx \Theta(n^{2.79512848736}) \end{aligned}$$

(a) 70 × 70:

$$\begin{aligned} T(n) &= 143640 \cdot t(\frac{n}{70}) + n^2 \\ &= \Theta(n^{\log_{70} 143640}) \\ &\approx \Theta(n^{2.79512268975}) \end{aligned}$$

(a) 72 × 72:

$$\begin{aligned} T(n) &= 155424 \cdot t(\frac{n}{72}) + n^2 \\ &= \Theta(n^{\log_{72} 155424}) \\ &\approx \Theta(n^{2.79514739109}) \end{aligned}$$

From the above run-times, the method of multiplying 70 × 70 matrices using 143,640 multiplications yields the best asymptotic running time and it is better than Strassen's algorithm whose running time $\approx \Theta(n^{2.8074})$

2 LUP decomposition

Problem statement:
Solve the equation by using an LUP decomposition.

$$\begin{pmatrix} 1 & 5 & 4 \\ 2 & 0 & 3 \\ 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 9 \\ 5 \end{pmatrix}$$

Solution:

$A = \begin{pmatrix} 1 & 5 & 4 \\ 2 & 0 & 3 \\ 5 & 8 & 2 \end{pmatrix}$ $B = \begin{pmatrix} 12 \\ 9 \\ 5 \end{pmatrix}$

LUP decomposition of A:

Swapping Row 3 with Row 1 as 5 is the pivot

$$\begin{aligned} A &= \begin{pmatrix} 5 & 8 & 2 \\ 2 & 0 & 3 \\ 1 & 5 & 4 \end{pmatrix} \\ &= \begin{pmatrix} 5 & 8 & 2 \\ 0.4 & -3.2 & 2.2 \\ 0.2 & 3.4 & 3.6 \end{pmatrix} \end{aligned}$$

Swapping Row 3 with Row 1 as 3.4 is the pivot

$$\begin{aligned} &= \begin{pmatrix} 5 & 8 & 2 \\ 0.2 & 3.4 & 3.6 \\ 0.4 & -3.2 & 2.2 \end{pmatrix} \\ &= \begin{pmatrix} 5 & 8 & 2 \\ 0.2 & 3.4 & 3.6 \\ 0.4 & -\frac{3.2}{3.4} & \frac{11.52}{3.4} + 2.2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0.2 & 1 & 0 \\ 0.4 & -\frac{3.2}{3.4} & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 8 & 2 \\ 0 & 3.4 & 3.6 \\ 0 & 0 & \frac{11.52}{3.4} + 2.2 \end{pmatrix} \\ &= L \cdot U \end{aligned}$$

Now, We have P as below based on the swapping of rows

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Forward substitution:

$$L \cdot y = P \cdot B$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0.2 & 1 & 0 \\ 0.4 & -\frac{3.2}{3.4} & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 12 \\ 9 \end{pmatrix}$$

Solving the above equation,

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \\ \frac{35.2}{3.4} + 7 \end{pmatrix}$$

Backward substitution:

$$U \cdot x = y$$

$$\begin{pmatrix} 5 & 8 & 2 \\ 0 & 3.4 & 3.6 \\ 0 & 0 & \frac{11.52}{3.4} + 2.2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \\ \frac{35.2}{3.4} + 7 \end{pmatrix}$$

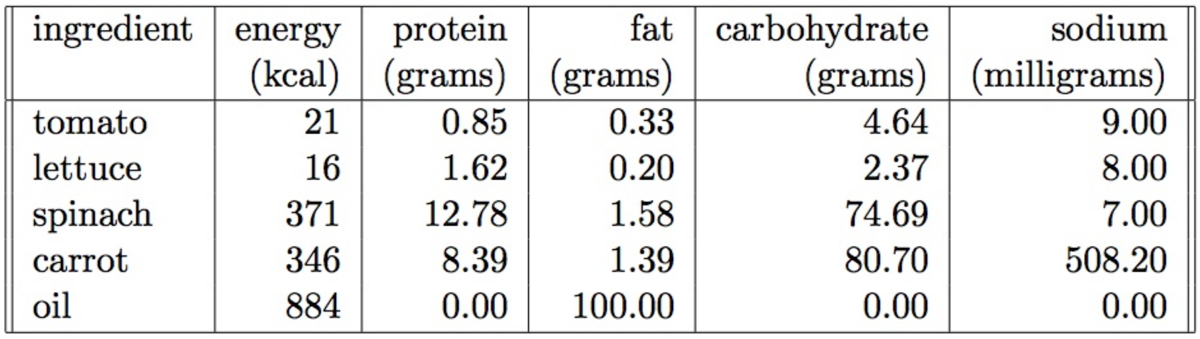
Solving the above equation,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -\frac{3}{19} \\ -\frac{1}{19} \\ \frac{59}{19} \end{pmatrix}$$

The values of the unknown variables x_1, x_2, x_3 are as below:
$x_1 = \frac{-3}{19}$, $x_2 = \frac{-1}{19}$, $x_3 = \frac{59}{19}$

3 Making a salad

Problem statement:
A salad is any combination of the following ingredients: (1) tomato, (2) lettuce, (3) spinach, (4) carrot, and (5) oil. Each salad must contain: (a) at least 15 grams of protein, (b) at least 2 and at most 6 grams of fat, (c) at least 4 grams of carbohydrates, (d) at most 100 milligrams of sodium. Furthermore, (E) you do not want your salad to be more than 50% greens by mass. The nutritional contents of these ingredients (per 100 grams) are



ingredient	energy (kcal)	protein (grams)	fat (grams)	carbohydrate (grams)	sodium (milligrams)
tomato	21	0.85	0.33	4.64	9.00
lettuce	16	1.62	0.20	2.37	8.00
spinach	371	12.78	1.58	74.69	7.00
carrot	346	8.39	1.39	80.70	508.20
oil	884	0.00	100.00	0.00	0.00

Find a linear programming applet on the Web and use it to make the salad with the fewest calories under the nutritional constraints. Describe your linear programming formulation and the optimal solution (the quantity of each ingredient and the value). Cite the Web resources that you used.

Solution:

Let the given information be represented as the below vectors. Let i_{th} ingredient be {1: tomato, 2: lettuce, 3: spinach, 4: carrot, 5: oil}

$\forall i \in I$, where $I = \{1, 2, 3, 4, 5\}$
$E_i \rightarrow$ Energy of the i^{th} ingredient in kcal per 100gms
$P_i \rightarrow$ Protein in grams of the i^{th} ingredient per 100gms
$F_i \rightarrow$ Fat in grams of the i^{th} ingredient in grams per 100gms

$C_i \rightarrow$ Carbohydrate in grams of the i^{th} ingredient in grams per 100gms
$S_i \rightarrow$ Sodium in grams of the i^{th} ingredient in grams per 100gms
$G_i \rightarrow$ Quantity of the i^{th} ingredient for an optimal solution in grams per 100gms

Expressing the given problem in terms of Linear program, we can find the optimal solution by minimizing the number of calories in the salad satisfying the constarints given as below:

Minimize: $\sum_{i \in I} G_i(E_i)$

Constraints: Each salad must contain

1. At-least 15 grams of protein: $\sum_{i \in I} G_i P_i \geq 1500$
2. At-least 2 grams of fat: $\sum_{i \in I} G_i F_i \geq 200$
3. Atmost 6 grams of fat: $\sum_{i \in I} G_i F_i \leq 600$
4. At-least 4 grams of carbs: $\sum_{i \in I} G_i C_i \geq 400$
5. Atmost 100 milligrams of sodium: $\sum_{i \in I} G_i S_i \leq 10$
6. Salad must include more than 50% greens by mass

$$0.5 \cdot (G_1 + G_2 + G_3 + G_4 + G_5) \geq G_2 + G_3$$
$$0.5 \cdot (G_1 + G_4 + G_5) - 0.5 \cdot (G_2 + G_3) \geq 0$$

The Simplex tool used to calculate the optimal objective value is available at
http://www.phpsimplex.com/simplex/simplex.htm?l=en

After entering the data provided in the question, below is the optimal solution

Number of calories: 232.51469899578 kcal

Quantity of ingredients in grams:

1. Tomato = 588.48014985171
2. Lettuce = 584.31760237265
3. Spinach = 4.1625474790567
4. Carrot = 0
5. Oil = 0

4 Minimum-cost multicommodity flow

Problem statement:
In the minimum-cost multicommodity-flow problem, we are given directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$ and a cost $a(u, v)$. As in the

multicommodity-flow problem, we are given k different commodities, $K_1, K_2, ...K_k$, where we specify commodity i by the triple $K_i = (s_i, t_i, d_i)$. We define the flow f_i for commodity i and the aggregate flow f_{uv} on edge (u, v) as in the multicommodity-flow problem. A feasible flow is one in which the aggregate flow on each edge (u, v) is no more than the capacity of edge (u, v). The cost of a flow is $\sum_{u,v \in V} a(u, v) f_{uv}$, and the goal is to find the feasible flow of minimum cost. Express this problem as a linear program.

Solution:

Based on the Multicommodity flow problem mentioned in $(CLRS,\ page\ 863)$, we can convert the problem in terms of linear program

Minimize: $\sum_{u,v \in E} a(u, v) f_{uv}$, where $f_{uv} = \sum_{i=1}^{k} f_{iuv}$

Constraints:

$$\sum_{i=1}^{k} f_{iuv} \leq c(u, v) \quad \forall u, v \in V$$

$$\sum_{v \in V} f_{iuv} - \sum_{v \in V} f_{ivu} = 0 \quad \forall i \in \{1, 2, ..., k\} \ and \ \forall u \in V - \{s_i, t_i\}$$

$$\sum_{v \in V} f_{i,s_i,v} = \sum_{v \in V} f_{i,v,t_i} = d_i \quad \forall i \in \{1, 2, ..., k\}$$

$$f_{iuv} \geq 0 \quad \forall u, v \in V \ and \ \forall i \in \{1, 2, ..., k\}$$