Laboratory practice No. 5: Divide to Conquer and Dynamic Programming

Miguel Sosa Villegas
Universidad Eafit
Medellín, Colombia
msosav@eafit.edu.co

Sergio A. Córdoba M.
Universidad Eafit
Medellín, Colombia
sacordobam@eafit.edu.co

Miguel Jaramillo Arenas
Universidad Eafit
Medellín, Colombia
mjaramil20@eafit.edu.co

3) Practice for final project defense presentation

3.1 The data structure we use was a matrix and the algorithm we implemented is a mixture between greedy and backtracking.
3.2 For a 50 node graph the minimum steps are 2500.
3.3 The data structure we used was a matrix and the algorithm was the travelling salesman problem algorithm. To know the distance you have to subtract coordenate “x” of point a and b and the same with the coordenate “y” and finally sum those two and there you have the distance between each radioactive waste. Then we translated the matrix into a graph and we ran the algorithm over that graph.
3.4 This point is the same as point 3.4.
3.5 The complexityO(n2*2n).
3.6 n is the number of radioactive waste.

4) Practice for midterms

4.1 *Optional*
4.2 ***LCS Backtracking***
- ***4.2.1 Complexity: O****(lenx * leny)*
- ***4.2.2*** *return table[i][j]*

4.3 *Optional*
4.4 *Optional*
4.5 Binary Search
- ***4.5.1 Complexity: c)*** ***T(n)*** *= T(n/2) + C que es O(log n)*
- ***4.5.2 Line number 8:*** *a[mitad]*
- ***4.5.3 Line number 15: (****a, mitad + 1, de, z)*

4.6 Maximum Increasing Subsequence
- ***4.6.1 Line number 7:*** *scm[i] = 1*
- ***4.6.2 Line number 12:*** *scm[i] = scm[j] + 1*
- ***4.6.3 Line number 16:*** *max = scm[i]*

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4.6.4 Complexity: c) O(n^2)

4.7 Floyd-Warshall Algorithm

4.7.1 Line number 12: *d[i][j]*

4.7.2 Line number 13: *d[k][j]*

4.7.3 Line number 14: *d[i][k]*

4.7.4 Complexity: O(n^3)

5) Recommended reading (optional)

Summary

Dynamic programming is based on optimization. The dynamic programming strategy it's a useful technique for solving many combinatorial optimization problems. Is based on decomposing the problem into subproblems, recursively applying the same method. This algorithm can be used when finding the shortest way is needed.

As well as branch-and-bound algorithms, dynamic programming allows us to avoid an exhaustive search in the solution space. Also, it is an elimination method where in every step, many solutions are eliminated.

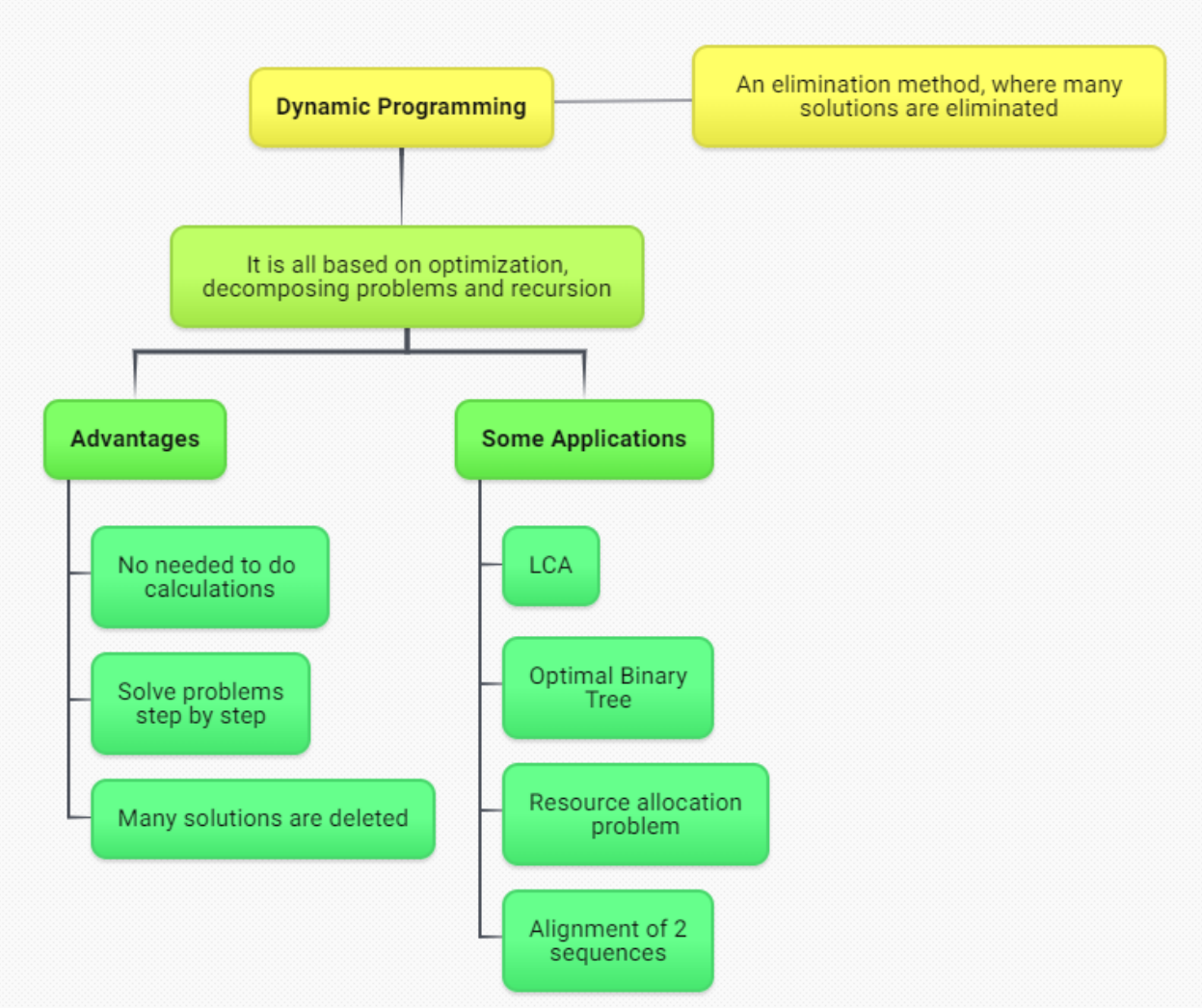
Advantages: many solutions can be eliminated and also save the trouble of making calculations. Finally, dynamic programming can be used to solve problems step by step (systematically).

Applications:

- Resource allocation problem
- Longest common subsequence
- Alignment of 2 sequences
- Maximum RNA pair mating
- Bag 0/1 problem
- Optimal binary tree
- Weighted perfect dominance in trees

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

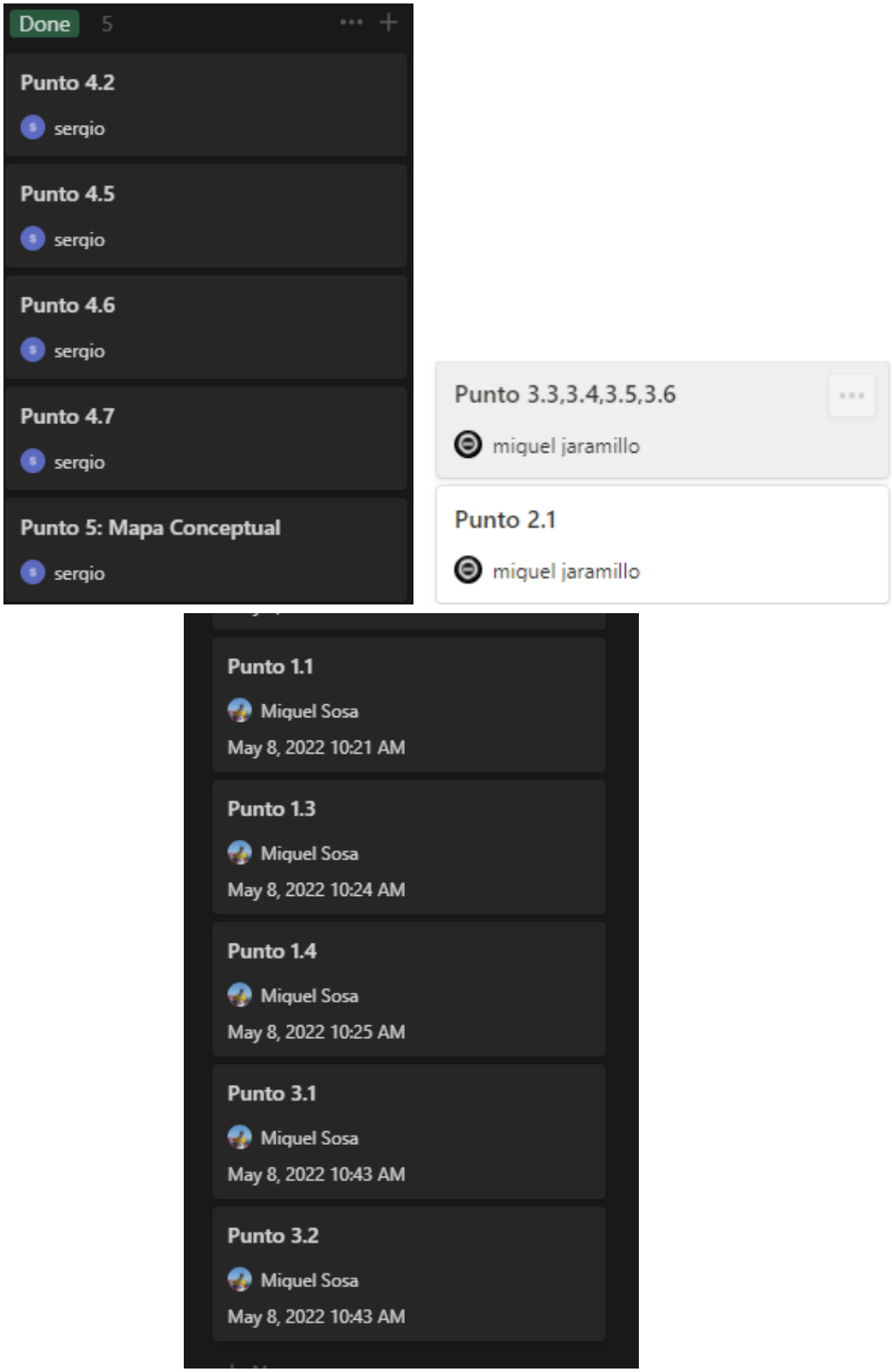
Conceptual Map



6) Teamwork and gradual progress (optional)

6.1 Kanban Board

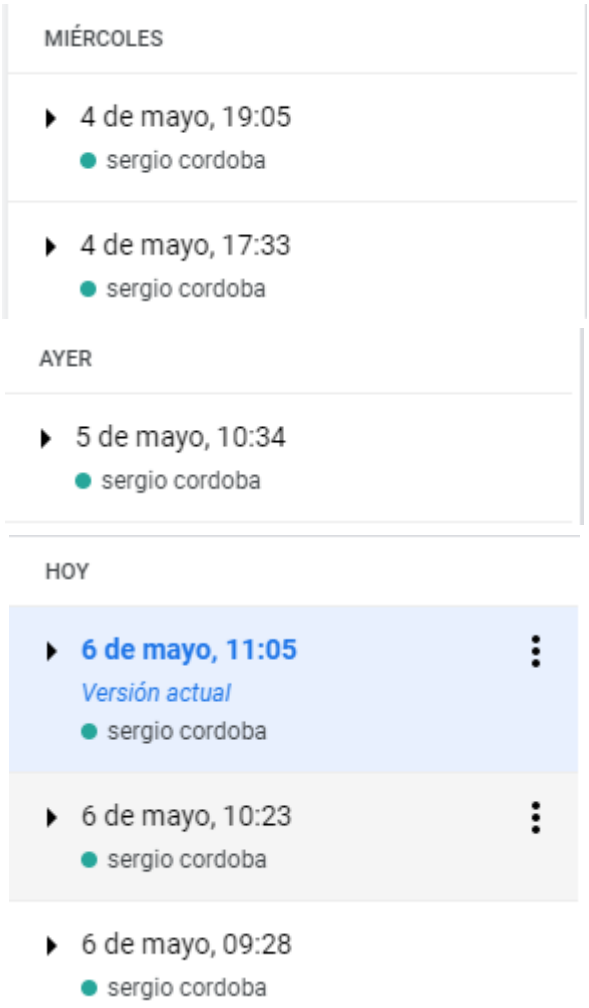
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



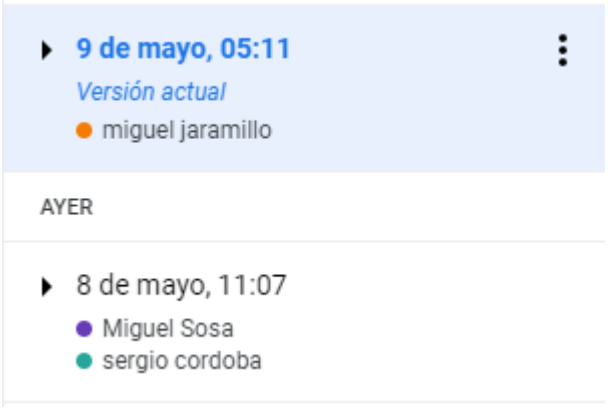
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

6.2 History of changes of the code

6.3 History of changes of the report



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473