Laboratory practice No. 5: Divide and conquer, dynamic programming

Yhoan Alejandro Guzmán García
Universidad Eafit
Medellín, Colombia
yaguzmang@eafit.edu.co

Juan Sebastian Pérez Salazar
Universidad Eafit
Medellín, Colombia
jsperezs@eafit.edu.co

3) Practice for final project defense presentation

3.1 The data structure used is an adjacency matrix, this due to the fact that the given graphs are always complete (when using HeldKarp).
The HeldKarp algorithm is a dynamic programming approach for the Traveling Salesman Problem, it starts by setting a start vertex and then calculating the power set with the remaining. Then it calculates and stores the cost from the start vertex to every other vertex, crossing all subsets of the power set. In the last iteration different possible paths are gotten, so the path with minimum cost is picked and that is the answer.

3.2 Since the complexity of the HeldKarp algorithm is $O(2^n n^2)$ (where n is the number of vertices), for n = 50 we would have to execute 2 814 749 767 106 560 000 operations.

3.3 The objective is not going through all the graph, since that would mean the whole city is being visited, so the objective is just to take the delivery nodes into account. For the distances, it is important to know that we are concerned about both spatial distance and time cost, because electricity is still consume when the electric car is stuck on traffic. For example, it doesn't matter if a path is only 2 kilometers if the car would have to go through it in an hour because of the traffic. So, the spatial distance can be easily calculated having the graph, but the cost in terms of time is harder to get since we don't have any traffic information, therefore we would have to use a service of an external company, like google.

3.4 The structure used to do this point is a matrix and lists, using Dynamic programming. The algorithm reads the file and stores the robot and garbage in a matrix. Then go through a list where it analyzes all the garbage and verify which is the best option to choose. Then, when choosing which garbage collects first, analyze which of the following garbage you must collect to reach the minimum path.

3.5 The complexity of the algorithm is O(n^2)

3.6 n means the number of trashes or garbages

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) Practice for midterms

4.1 4.1.1

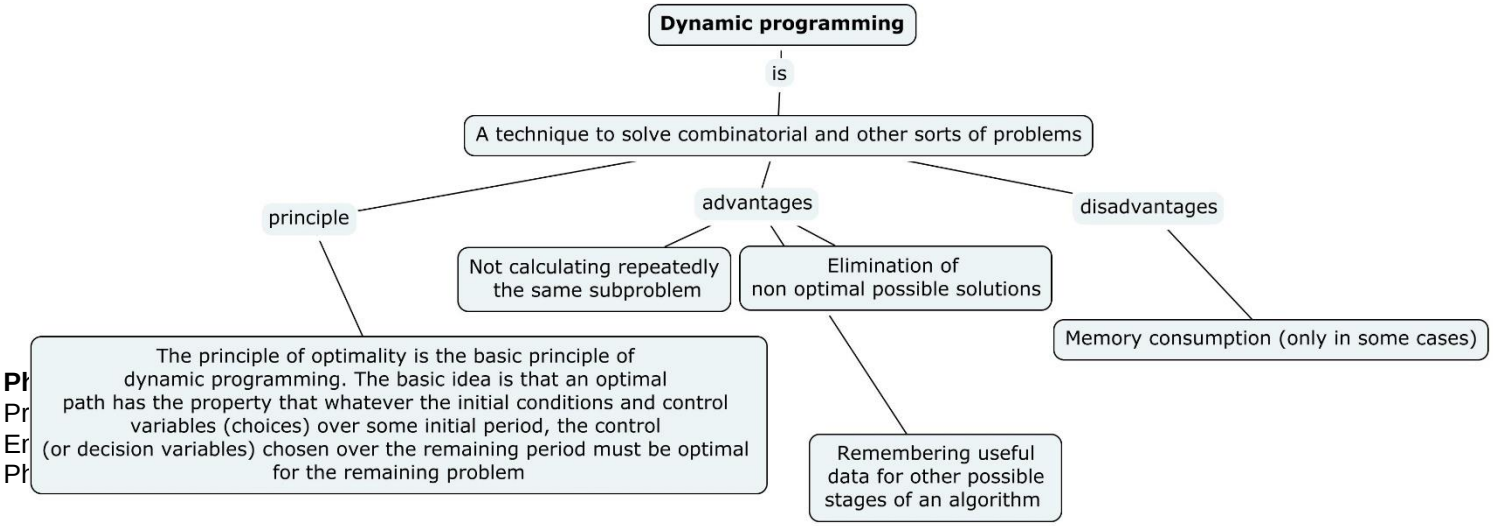
	C	A	L	L	E
C	0	1	2	3	4
A	1	0	1	2	3
S	2	1	1	2	3
A	3	2	2	2	3

4.1.2

	M	A	D	R	E
M	0	1	2	3	4
A	1	0	1	2	3
M	2	1	1	2	3
A	3	1	2	2	3

4.2 4.2.1 (lenx*leny)
4.2.2 return table[lenx][leny]
4.3 4.3.1 a) O(n)
4.3.2 a) T(n) = c1:n + c2
4.4 length-1
4.5 4.5.1 c) T(n) =T(n/2) + C that is O(log n)
4.5.2 return a[mitad]
4.5.3 return bus(a, mitad +1, de, z)
4.6 4.6.1 scm[i] = 0
4.6.2 scm[i] = scm[i] +1
4.6.3 max = scm[i]
4.6.4 c) $O(n^2)$
4.7 4.7.1 d[i][j]
4.7.2 d[i][k]
4.7.3 d[j][k]
4.7.4 $O(n^3)$

5) Recommended reading (optional)



6) Team work and gradual progress (optional)

6.1 Meeting minutes

Member	Date	Done	Doing	To do
Yhoan	5/05/2019	**Point 1.1**		test for points 1.1
Yhoan	5/05/2019	**test for point 1.1**	a better solution	Practice for final project defense presentation
Sebastián	5/05/2019	**Practice for final project defense presentation**	Practice for midterms	point 1.1
Sebastián	5/05/2019	**Practice for midterms**	point 2.1	
Sebastián	5/05/2019	**Point 2.1**		test for point 2.1
Sebastián	5/05/2019	**test for point 2.1**		points of part 3
Yhoan	4/05/2019	**points of part 3**		recommended reading
Yhoan	4/05/2019	**recommended reading**		upload the laboratory

6.2 History of changes of the code

History changes of code		
Version	Code	Status
1.0	1.1	
2.0	1.1	
1.0	2.1	
2.0	2.1	

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