ASSIGNMENT 5: GREEDY ALGORITHM

due date: Sunday, April 17th 2022 (23.59 WITA)

Aturan pengerjaan tugas:

1. Kerjakan **semua** soal yang ada secara singkat, padat, dan jelas.
2. Tugas boleh diketik/ditulis tangan (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah. Jika menggunakan tulis tangan, harap discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file. Tulis jawaban pada satu file pdf
3. Format penamaan tugas: **NamaLengkap_Kelas_NIM**. Contoh: **GedeGanesha_6A_1610101001.pdf**. Pengumpulan tugas melalui e-learning Undiksha.
4. Anda diizinkan untuk berdiskusi dengan rekan Anda. Namun Anda harus menuliskan/menjelaskan jawaban Anda sendiri, dan paham dengan baik apa yang Anda tulis. Anda siap bertanggung jawab terhadap hasil pekerjaan Anda. Hasil pekerjaan yang memiliki kemiripan yang tinggi dengan pekerjaan mahasiswa lain mempengaruhi poin penilaian.
5. Tugas dinilai berdasarkan kerapian penulisan, dan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.

1 Job scheduling with deadlines

Given n jobs that will be done by a machine. Each job is processed by the machine in one unit time and deadline of every job i is $d_i \geq 0$. Job i will give profit p_i if and only if the job is done not after the deadline. How to choose the jobs that will be processed by the machine so that the profit is maximum?

1. Given 4 jobs ($n = 4$) with the following characteristics:
   - $(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$
   - $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

   Let the machine starts to work at 6 am, then we have the following constraints:

Job	**Deadline (d_i)**	**Must be done before**
1	2 hours	8 am
2	1 hour	7 am
3	2 hours	8 am
4	1 hour	7 am

   Let J be the set of jobs, then the objective function of this problem is:

   $$\text{Maximize } F = \sum_{i \in J} p_i$$

   - A solution set J is *feasible* if every job in J is done before the deadline.
   - An *optimal solution* is a feasible solution that maximize F.

   TASK: Solve the problem with exhaustive search. For this, create a table containing, set of solutions, order of processing, total profit, and feasibility.

   - *Set of solution*: possible subset of chosen jobs
   - *Order of processing*: the order of job processed so that the deadline is respected.
   - *Total profit*: total of profit based on the set of feasible solutions
   - *Feasibility*: whether the set of solution is feasible/not feasible

Set of jobs	**Order**	**Total profit (F)**	**Description**
$\{\}$	-	0	feasible
$\{1\}$	1	50	feasible
$\vdots$	$\vdots$	$\vdots$	$\vdots$

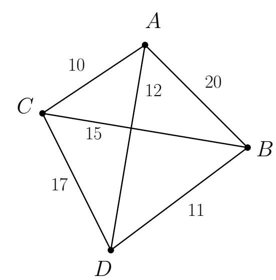
2. What is the complexity of the exhaustive search approach?
3. Consider the following Greedy strategy for the probem. To choose a job: "at each step, choose job i with the largest p_i to increase the objective value F". Solve the problem with the Greedy strategy and find the *optimal solution* and the *optimal profit*. (The optimal solution is a solution set that gives the optimal profit.)
4. Write a pseudocode to implement the Greedy strategy above!
5. What is the complexity of the Greedy algorithm?
6. Explain how to improve the Greedy algorithm above?

2 Traveling salesman problem

TSP: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Let the vertices of the input graph G be: $v_1, v_2 \ldots, v_n$. Let the tour is started from v_1. The next vertex is chosen "greedily": At each step i, choose the vertex v_j (among the available vertices) for which the weight of edge (v_i, v_j) is minimized.

Given the following graph. Implement the Greedy algorithm to find a TSP solution for this graph. Write the TSP solution and compute the total weight of the solution.



3 Knapsack problem

1. **Integer knapsack problem**

   Recall that in the Integer (1/0) Knapsack Problem, there are three greedy strategies that can be implemented to obtain a solution for the knapsack problem: Greedy by profit, Greedy by weight, and Greedy by density. However, it is claimed that none of the strategies guaranteed an optimal solution.

   Give an instance of the Integer Knapsack Problem, where the optimal solution is different than the solution obtained by implementing the three Greedy strategies (by profit, by weight, and by density).

   For this question, you should give an instance, the solution of the three strategies (as explained in the lecture), and the optimal solution (give a proof/an argument to show that it is indeed an optimal solution). You are not allowed to give the same example as your fellow.

2. **Fractional knapsack problem**

   The following theorem is given in the lecture:

   Theorem 3.1. *If $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$, then the greedy algorithm with the strategy of choosing the maximum $\frac{p_i}{w_i}$ gives an optimal solution.*

   Give a formal proof of the theorem!