ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

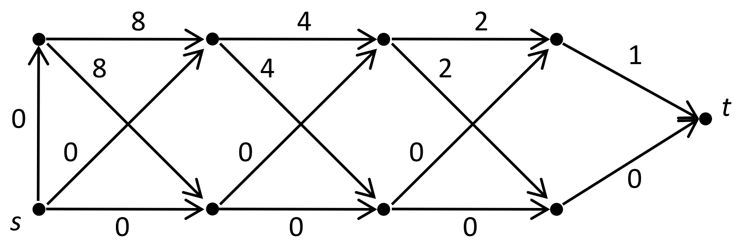
Bachelor semester 4-DO

DISCRETE OPTIMIZATION WEEK 12

EXERCISE 1

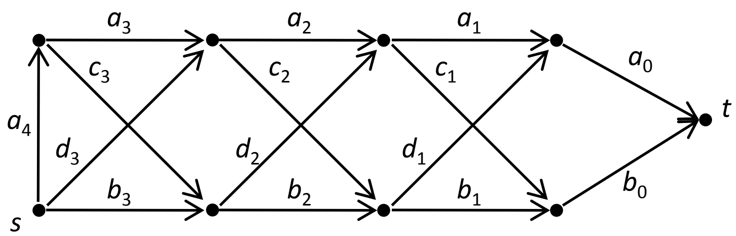
The objective of this exercise is to show that Ford's algorithm can take an exponential number of steps in the worst case.

Consider the following directed graph :



Show that there is a sequence of steps in which the variable y_t of the node t takes all the values 15, 14, 13, ..., 1, 0.

Solution :



After correcting the edges a_4, a_3, a_2, a_1, a_0 we have $y_t = 15$.
After correcting c_1, b_0 we have $y_t = 14$.
After correcting c_2, d_1, a_0 we have $y_t = 13$.
After correcting b_1, b_0 we have $y_t = 12$.
After correcting c_3, d_2, a_1, a_0 we have $y_t = 11$.
After correcting c_1, b_0 we have $y_t = 10$.
After correcting b_2, d_1, a_0 we have $y_t = 9$.
After correcting b_1, b_0 we have $y_t = 8$.
After correcting d_3, a_2, a_1, a_0 we have $y_t = 7$.
After correcting c_1, b_0 we have $y_t = 6$.
After correcting c_2, d_1, a_0 we have $y_t = 5$.
After correcting b_1, b_0 we have $y_t = 4$.
After correcting b_3, d_2, a_1, a_0 we have $y_t = 3$.

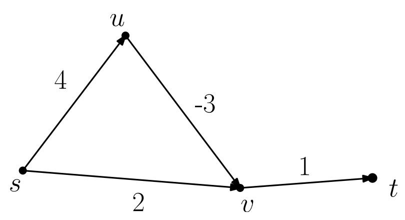
After correcting c_1, b_0 we have $y_t = 2$.
After correcting b_2, d_1, a_0 we have $y_t = 1$.
After correcting b_1, b_0 we have $y_t = 0$.

The above graph can be extended in the natural way, yielding for every n a graph of size linear in n, but for which Ford's algorithm takes roughly 2^n steps in the worst case.

<u>EXERCISE 2</u>

Give an example to show that Dijkstra's algorithm can give incorrect results if negative costs are allowed.

Solution :



Let s be the source in the graph (above), i.e. a vertex from which we are computing all the distances. Dijkstra's algorithm computes the distance from s to t do be 3. Indeed, the first node besides s we visit is v and we set its distance from s to be 2. Then we visit t and set its distance from s to be 3. Finally, we visit u and set its distance from s to be 4. Then we change the distance to v to be 1, but we never change the distance to t again.