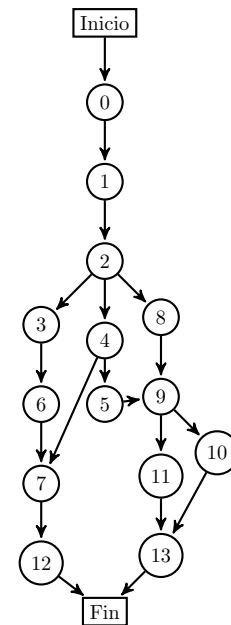


## *Simulación y estimación de proyectos*

### Simulación y duración de proyectos

La empresa para la cual trabajamos ha ganado una licitación para un proyecto de infraestructura por 5.000.000 de dólares. El mismo está compuesto por una serie de tareas, con dependencias entre sí. Para cada tarea se tienen además tres estimaciones respecto a la duración: una estimación *optimista* (op), la *más probable* (ml) y la *pesimista* (ps). La duración total del proyecto depende de la duración individual de cada tarea, que *a priori* no conocemos, y de las dependencias entre ellas. El listado de las tareas junto con el grafo de dependencias pueden verse en las figuras debajo. En este caso en particular, el grafo de tarea contiene 6 posibles caminos, y para estimar la duración total del proyecto alcanza con estimar la duración de cada posible camino y tomar el máximo.

#Tarea	Nombre	Estimación		
		op	ml	ps
0	Excavación	1.0	2.0	3.0
1	Cimientos	2.0	3.5	8.0
2	Pared gruesa	6.0	9.0	18.0
3	Techo	4.0	5.5	10.0
4	Cañerías exteriores	1.0	4.5	5.0
5	Cañerías interiores	4.0	4.0	10.0
6	Revestimiento exterior	5.0	6.5	11.0
7	Pintura exterior	5.0	8.0	17.0
8	Electricidad	3.0	7.5	9.0
9	Tableros	3.0	9.0	9.0
10	Pisos	4.0	4.0	4.0
11	Pintura interior	1.0	5.5	7.0
12	Arreglos exteriores	1.0	2.0	3.0
13	Arreglos interiores	5.0	5.5	9.0



La licitación contempla un plazo de ejecución para del proyecto de 47 semanas. Si el proyecto se extiende de este plazo, hay una penalización de 300.000 dólares. Por el contrario, existe un incentivo de 150.000 dólares si el proyecto es finalizado en la semana 40 o antes.

La empresa se encuentra considerando invertir 100.000 dólares para contrar mano de obra extra y mejorar el peor escenario de duración (ps) de la tarea 2, bajándolo de 18 semanas a 13. Sin embargo, la gerencia a cargo de la ejecución no se encuentra convencida respecto a la conveniencia de realizar (o no) la inversión.

### El contexto

Contamos con un desarrollo parcial para nuestro problema, que incluye la clase **ProjectGraph** con la información correspondiente a cada tarea y los 6 posibles caminos en el grafo de dependencias. El objetivo es utilizar técnicas de simulación para obtener estimadores de las

probabilidades de ciertos escenarios que, luego, serán incorporados a un árbol de decisión y finalmente determinar si conviene (o no) contratar recursos extra.

La clase `ProjectGraph` contiene información estática (no se levanta de archivos). Para este proyecto, cuenta con las siguiente estructuras:

- `n (int)`: cantidad total de tareas. Las tareas se enumeran de 0 a  $n - 1$  (en particular, 0 a 13).
- `m (int)`: cantidad total de caminos.
- `paths (list(list(int)))`: lista de 6 posiciones, con `paths[i]` el  $i$ -ésimo camino. Un camino se representa como una secuencia de tareas.
- `dist_params (dict<int, tuple>)`: un diccionario que contiene en `dist_params[i]` una tupla `t` de tres elementos como significado. En particular, `t[0] = op`, `t[1] = ml`, `t[2] = ps`.

## El trabajo

La resolución del trabajo consiste en las siguiente etapas, incluyendo completar el código del archivo `pert.simulation.py` completando y agregando aquellas funciones auxiliares que consideren necesarias. Se pide:

1. **(2 puntos) Árbol de decisión.** Modelar la decisión mediante un árbol que contemple todos los escenarios posibles. Completar los valores que se encuentren disponibles y mencionar aquellos que necesiten ser calculados.
2. **(4 puntos) Simulación de duración de proyecto.** Utilizar Python para realizar 1000 simulaciones, dadas las definiciones de las duraciones de las tareas, implementando las funciones que se describen a continuación y retornar los resultados para su posterior análisis. En particular, considerar:
  - `simulate_triangular()`: toma tres parámetros con los valores `op`, `ml` y `ps` y genera un número aleatorio con distribución triangular. Pueden simular usando `scipy.stats.triang` o `numpy.random.triangular`.
  - `simulate_tasks_duration()`: toma el grafo y devuelve una lista `ret` de  $n$  elementos, con `ret[i]` el tiempo (aleatorio) de la tarea  $i$  generado usando los parámetros correspondientes.<sup>1</sup>
  - `get_path_duration()`: toma un camino, `path`, y una lista con las duraciones de las tareas, y devuelve el tiempo de finalización de ese camino.
  - `get_project_duration()`: toma una instancia de `ProjectGraph`, `project_graph`, y una lista con la duración de cada tarea y devuelve la duración total del proyecto. (Recordar que es el máximo de las duraciones individuales de cada camino en `project_graph.paths`.<sup>2</sup>

---

<sup>1</sup>Notar que debe llamar a `simulate_triangular()`

<sup>2</sup>Notar que debe llamar a `get_path_duration()`

- `simulate()`: toma un número de simulaciones, `n_sim` y el grafo del proyecto y realiza `n_sim` simulaciones de ejecución del proyecto. Retorna una lista de `n_sim` elementos con la duración de cada simulación.<sup>3</sup>
3. **(2 puntos) Cálculo de probabilidades e interpretación de resultados.** Considerar los dos escenarios posibles, **con** y **sin** contratación adicional para la tarea 2 y calcular las probabilidades de terminar en la semana 40 o antes, entre la semana 40 y 47, o luego de la semana 47. Para esto último, completar y utilizar la función `get_probs_in_range()`, que toma una lista de valores, `vals`, dos valores `a` y `b`, y devuelve la proporción de elementos en `vals` en  $(a, b]$ . Para considerar ambos escenarios, en la función `main()` se considera modificar convenientemente la información de la tarea dos y así reflejar el cambio en `ps` para la tarea 2.
  4. **(2 puntos) Resolución del árbol y decisión final.** Finalmente, con las probabilidades estimadas resolver el árbol de decisión planteado y sugerir la decisión a tomar por la compañía.

### Modalidad de entrega

Junto a cada uno de los ejercicios se les pide que presenten un informe de máximo 5 páginas que contenga:

- introducción al problema,
- descripción del modelo propuesto, según corresponda,
- consideraciones generales respecto a la implementación del modelo, incluyendo dificultades que hayan encontrado,
- resumen de resultados obtenidos en la experimentación,
- conclusiones, posibles mejoras y observaciones adicionales que consideren pertinentes.

Junto con el informe debe entregarse el código con la implementación del modelo. El mismo debe ser entendible, incluyendo comentarios que faciliten su corrección y ejecución.

### Fechas de entrega

**Formato Electrónico:** **20 de Junio de 2021, 23:59 hs**, enviando el trabajo (informe + código) a las dirección `dm_mim@utdt.edu`. El subject del email debe comenzar con el texto **Trabajo Práctico 1 – Modelos de Decisión** seguido de la lista de apellidos de los integrantes del grupo. Todos los integrantes del grupo deben estar copiados en el email.

**Importante:** El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.

---

<sup>3</sup>Notar que debe llamar a `get_project_duration()` y `simulate_tasks_duration()` y combinarlas.