

Tarea N° 2
Sistemas Operativos

Fecha: Jueves, 8 de Mayo de 2025

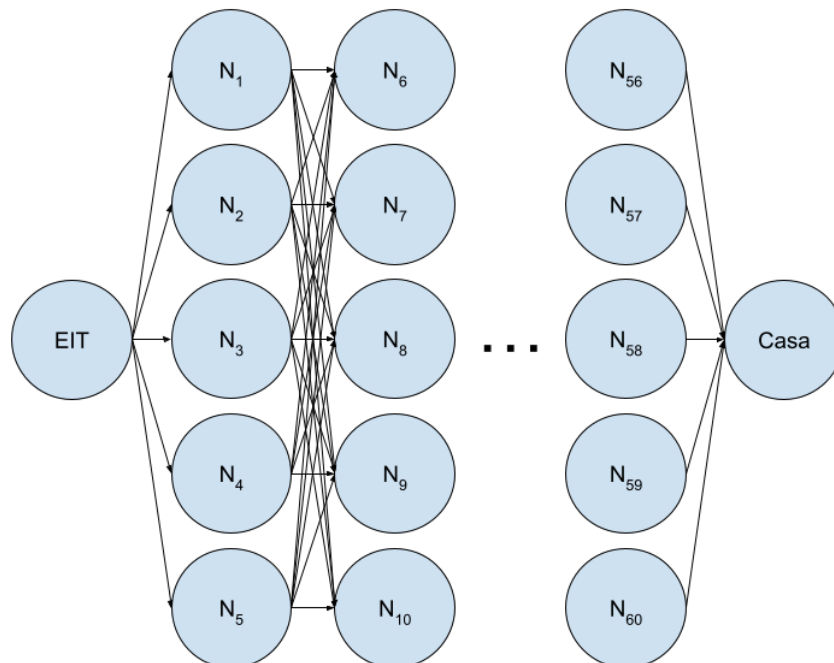
Plazo: 3 semanas

Uno de los profes de SO está ansioso de aprovechar mejor su (poco...MUY poco) tiempo libre para jugar a la nueva Nintendo Switch 2, es por eso que quiere encontrar una buena ruta para ir desde la EIT a su casa. Sin embargo, debido a sus múltiples responsabilidades, que incluyen revisar controles, preparar exámenes, escribir papers... no dispone del tiempo suficiente para explorar manualmente todas las posibles rutas. Por lo tanto, he decidido delegar esta tarea a usted.

Para lograr esta tarea, se le ocurre al profe utilizar threads, para ir recorriendo cada una de las rutas sobre el grafo que conecta la EIT con la casa. Al finalizar el recorrido, se deberá registrar cuál fue la ruta y su costo total. Así entonces, deberá obtener cuál es la mejor ruta y cuánto costó.

Se pide:

- 1) **(5 puntos)** Diseñar un grafo acíclico dirigido (DAG) con 62 nodos, donde cada conexión entre nodos tenga un costo asociado. Este DAG debe seguir la siguiente estructura:



Notar que entre cada nivel del DAG, los nodos deberán estar completamente conectados entre sí. Cada una de las aristas deberá contener costos aleatorios enteros (entre 5 y 20).

- 2) **(35 puntos)** Implementar K threads (donde K puede ser 1, 10, 20, 50 o 100) que recorran las distintas rutas del DAG entre la casa del profesor y la EIT. Cada thread tomará decisiones aleatorias para seleccionar su próximo destino en cada paso. Lo anterior se deberá ejecutar durante un tiempo de 60 segundos, en donde cada vez que se encuentre una mejor ruta, el programa deberá reportar dicho costo. Todo esto debe realizarse de manera **sincronizada**.
- 3) **(20 puntos)** Implemente una versión en donde exista una limitación en la cantidad de threads que pueden transitar simultáneamente por una misma arista del grafo. Esta limitación está definida por un número L (donde L puede ser 2 o 3). Se requiere implementar herramientas de sincronización que cumplan con esta restricción. Realice los mismos experimentos del apartado anterior.

Además del código, deberá presentar un informe explicando su propuesta y los resultados experimentales obtenidos en cada caso. Este informe deberá contener gráficos que ilustren la evolución de la solución en cada escenario, mostrando, por ejemplo, la relación entre el tiempo de ejecución y la calidad de la solución alcanzada.

Asimismo, será necesario incluir una explicación y un análisis crítico de estos resultados, destacando patrones observados, posibles causas de variaciones en el desempeño.

OBSERVACIÓN:

Para la resolución de los problemas debe usar herramientas de sincronización vistas en clases, cualquier otro método utilizado no será considerado y no tendrá puntaje. Por supuesto, también se dará la directiva de comprobar contra soluciones construidas por chatbots generativos. Estas soluciones tendrán el puntaje mínimo.

Condiciones de entrega

Se indican las siguientes condiciones para la entrega de la tarea:

- La tarea se podrá desarrollar de manera **individual o en parejas**. Los códigos serán sometidos a comprobación automática de plagio y revisados contra códigos online ya existentes. Estos estarán escritos en C/C++ y usarán únicamente las librerías estándar.
- Además del código y el informe, deberá incluir un archivo README que explique cómo se debe ejecutar su código.
- Es requisito indispensable que los códigos compilen. De no cumplirse esta condición, la tarea tendrá un descuento inicial de 20 puntos.
- La tarea se entregará vía el Canvas del curso. La fecha y hora límite para la entrega es el 1 de Junio a las 23:59. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato. **No habrá extensión de plazo.**