

Programación Dinámica (Parte 2)

El grafo que se usó para el desarrollo de esta práctica es la Figura 1, aunque también viene anexo al correo, cumple con las condiciones que se comentaron en la práctica pasada.

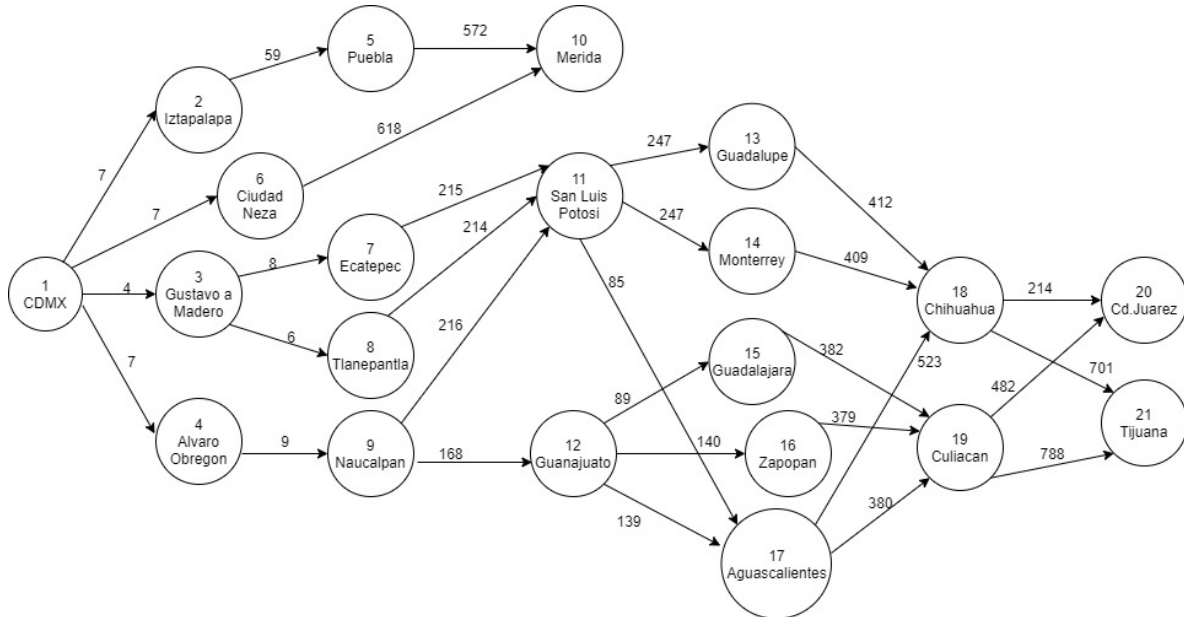


Figura 1. Grafo de CDMX a diferentes estados

Para poder implementar la recursividad hacia ATRÁS y ADELANTE se optó por usar una matriz de adyacencia, que como se comentó en la práctica pasada, esta matriz contendrá el valor del peso en donde un nodo se conecta con otro directamente, esto se hace ya que es necesario tomar el subproblema final (nodo final) y regresar al subproblema inicial (nodo inicial), es decir, se sigue respetando la ponderación del grafo de la Figura 1, provocando así tener las mismas condiciones de grafos que en la práctica anterior se mencionaron.

Como conclusión la programación dinámica es muy interesante ya que permite solucionar un problema de dos diferentes que tienen el mismo resultado, pero una es más eficiente que la otra, al inicio fue algo confuso la recursión hacia ATRÁS pero tras leer un poco el documento y hacer investigación sobre el tema la recursión hacia ATRÁS fue más clara, al hacer la investigación también me topo de muchos más ejemplos de uso para la programación dinámica, lo más interesante es que la mayoría de esos ejemplos son los que ocasiones se ocupan para concursos de algoritmia.