

Exercici 4 (2 punts) (Discurs). Davant el gran nombre de intervencions sobre la utilitat de l'ensenyament universitari, el rector de la UPC ha decidit encarregar a UPCnet un mecanisme per redactar aquest discursos.

UPCnet ha desenvolupat un model consistent en un graf dirigit $G = (V, E, \omega)$, on cada aresta és etiquetada amb una paraula, i un vèrtex distingit $t \in V$. Segons aquest model, un camí al graf que finalitza a t representa una frase amb sentit que es pot incloure al discurs. La frase es correspon amb la seqüència ordenada de mots que apareixien al camí, seguint l'ordre establert pel propi camí. Abans de preparar un discurs el rector decidirà els punts d'inici del seu discurs, tot indicant una seqüència de nodes del graf on han de començar les frases que el formaran.

Per tal d'eliminar redundàncies al discurs, l'equip rectoral aconsella que el discurs mai faci servir més d'una vegada una aresta del graf.

- (a) Dissenyeu un algorisme per escriure un discurs que compleixi les especificacions del rector i segueixi els consells de l'equip rectoral. En cas que no fos possible, s'ha de presentar una versió reduïda del discurs on també s'especifiquin els nodes d'inici de frase que no s'han pogut completar.
- (b) Indiqueu com modificaríeu l'algorisme proposat a l'apartat previ si l'equip rectoral relaxa la condició prèvia de redundància i permet un nombre màxim c d'aparicions de cada aresta al discurs.

Sol.

Consideramos una red de flujo en la que añadimos al grafo un nuevo nodo s conectado con los nodos en la lista del rector. Como puede haber nodos repetidos, asociaremos a cada arista el número de apariciones del nodo en la selección del rector. El vértice t actuará como sumidero, ya que las frases acaban en t .

Las capacidades del resto de aristas son dependen del apartado.

- (a) Capacidad uno en todas las aristas, tal y como hemos visto en clase esto garantiza que cada arista se utiliza como mucho 1 vez que es lo que aconseja el equipo rectoral.

Una unidad de flujo en esta red representa un camino de s a t y permite generar una frase que se inicia en uno de los nodos seleccionado por el rector.

Si el valor del flujo máximo coincide con la longitud de la secuencia del rector, podemos proporcionar un discurso completo. En caso contrario las arista que salen de s y que no están saturadas nos proporcionan los nodos de inicio en los que no se ha podido completar frase.

Una vez obtenido un flujo f con valor máximo, iremos extrayendo uno a uno caminos de s a t en el grafo formado con las aristas e con $f(e) > 0$. Una vez extraído un camino reduciremos en una unidad el flujo en las aristas del camino y repetiremos el proceso hasta que tengamos todas las aristas con flujo 0.

El valor máximo del flujo es la longitud de la secuencia del rector, supongamos que la lista tiene k entradas. Los caminos de s a t siempre contienen una arista de capacidad 1, por lo que cada paso incrementamos el flujo en una unidad en cada ciclo. Esto nos da un coste de $k(n + m)$ para calcular un flujo con valor máximo.

Comprobar el flujo en las aristas que salen de s tiene coste $O(\min k, n)$ y extraer los caminos el mismo que obtener el Max Flow. Así el coste total del algoritmo es $O(k(n + m))$.

- (b) Aquí cada arista del grafo original se puede utilizar c veces, esto lo podemos conseguir asignando capacidad c .

El análisis es similar al caso anterior, como el valor del flujo máximo sigue acotado por k el coste total del algoritmo es $O(k(n + m))$.