

Especificaciones del proyecto

Deberan implementar Dinic, y programas auxiliares. Los programas deben ser en C.

Programa Principal

Input: El programa debe leer archivos que tendra lineas de la forma:

$xy\ c$

que representaran al lado \vec{xy} de capacidad c .

Output: El programa debera tener dos modos: verboso y no verboso. En el modo no verboso, solo se debe producir un archivo de salida que diga:

“Valor del flujo: #”

en una linea, donde # será el valor del flujo. En otra linea debe decir:

“Corte Minimal: $S=\{\dots\}$ ”

(los puntos indican vertices que deben ser escritos por el programa).

En el modo verboso, ademas de estas dos lineas, que iran AL FINAL del archivo, debe haber lineas anteriores de la forma:

$tzyx\dots as: \#$

que indicaran que se esta mandando # unidades de flujo a lo largo del camino $sa\dots zt$.

Si el camino usa lados backwards (del network original, obvio) se deben marcar usando “>”. Por ejemplo, el camino

$$s\overleftarrow{A}B\overleftarrow{C}D\overleftarrow{E}G\overleftarrow{I}K\overleftarrow{H}t : 7$$

sera escrito con la linea:

$tH>KIGEDC>BAs: 7$

Programas Auxiliares

1) Deben hacer un programa que genere archivos input al azar, cuando se les den parametros n y m . (ie, n vertices y m lados). Puede suponer que $n \leq 50$. Los nombres de los vertices deben ser (ademas de s y t) A, B, \dots, Z en ese orden, y si $n \geq 28$, seguir con a, \dots, z , en ese orden, pero por supuesto sin la s ni la t , que ya se usan. (no usar \tilde{N} ni \tilde{n})

2) Deben hacer un programa que, dados parametros n, m, k , genere k archivos de input con n vertices y m lados (no es necesario crearlos a todos y guardarlos, basta con crear de a uno por vez, usarlo y luego borrarlo), luego corra su programa de Dinic en cada archivo (en modo no verboso), luego registre el tiempo de cada corrida, tanto en clocks de la maquina como en microsegundos, y luego cree un archivo en el cual se escriba:

“Resultado de correr Dinic k veces en networks de n vertices y m lados:”

en la primera linea, seguida de:

“ Promedio: # clocks, # microsegundos

Minimo: # clocks, # microsegundos

Maximo: # clocks, # microsegundos

”

donde en promedio pondran el promedio de las corridas, y en minimo y maximo el minimo y el maximo.

El programa de testeo debe ser lo suficientemente general para que se pueda llamar a cualquier programa compatible con estas especificaciones, no solo el suyo propio. La medicion del tiempo no debe incluir el tiempo gastado en generar los archivos, solo el tiempo que tarda Dinic. Los usare a unos con otros, el grupo que tenga el algoritmo mas rapido (que sea correcto, obvio) ganara un punto extra en el final (todos los integrantes ganaran un punto extra), SIEMPRE que rindan en las fechas de Junio-Julio-Agosto.

Si dos o mas equipos empatan en el mas veloz, el punto se divide por la cantidad de equipos. (por ejemplo, si dos equipos empatan, todos los miembros de los dos equipos reciben medio punto en el final).

El programa principal puede suponer que no habra lados xy junto con lados yx . El programa que genere los archivos debe asegurarse que esto no pase.