Teoritävling 1

Regler

Inlämning sker främst via email till `johan.sannemo@kodsport.se` senast den 16 februari. Skriv helst digitalt och lämna in som en PDF. Om du skriver dina lösningar för hand ska du utöver att skicka in fotokopior över mail posta dina lösningar till

`Johan Sannemo`
`Evenemangsgatan 18, Lgh 2008`
`169 56 Solna`

För samtliga uppgifter behöver lösningen endast beskrivas teoretiskt, utan att implementeras i kod. Inkludera gärna pseudokod om du tycker det är lättare att beskriva än i text.

Deluppgifter klarar du genom att beskriva en lösning som är tillräckligt snabb. Vi bryr oss endast om tidskomplexiteten av din lösning som funktion av indatan, **om inte annat anges**. När du beräknar tidskomplexiteten är vi bara intresserade av den ledande termen. Detta betyder att en lösning som är t.ex. $O(N + M)$ med $N \leq M$ räknas som $O(M)$. Given den slutgiltiga tidskomplexiteten kommer vi beräkna uttrycket (alla logaritmer är i bas 2!) och kolla om det är mindre än 10^9 när vi avgör om du klarar en deluppgift eller inte. Delpoäng ges inte inom en deluppgift om du bara är lite över gränsen. Har du frågor kring om en viss tidskomplexitet är okej eller inte, *fråga Johan eller Mårten privat* på Discord eller via mailen ovan.

Tidskomplexitet krävs inte att du beräknar, även om vi uppskattar att du själv försöker beskriva den för rättandets och träningens skull. Felaktiga beskrivningar av tidskomplexitet ger inget avdrag.

Bevis krävs inte, även om du gärna får inkludera det, återigen för att rättandet ska bli enklare och för din egen övning. Detta innebär t.ex att du kan göra helt omotiverade antaganden utan att bevisa dessa som underlag för din lösning. Felaktiga bevis ger inte avdrag – du tjänar alltså inget på att *inte* försöka bevisa saker. Tvärtom kommer vi ge dig feedback på dina felaktiga bevis.

Avdrag kan komma att ges under flera omständigheter, t.ex. ofullständiga beskrivningar, specialfall du missar eller olika buggar.

Problem 1. Givet är ett $N \times M$ rutnät där varje ruta innehåller ett heltal. Givet är också Q instruktioner där varje instruktion ersätter ett tal i en ruta med ett större tal. Talen i rutorna är alltid mellan -10^9 och 10^9.

Beskriv en algoritm som efter varje instruktion skriver ut hur många rutor det finns i rutnätet som uppfyller att talet som står i rutan både är större än alla andra tal som står i samma rad och alla andra tal som står i samma kolumn. Det är garanterat att inga två rutor i rutnätet någonsin kommer innehålla samma tal.

Deluppgift 1 (2 poäng): $N, M, Q \leq 100$.
Deluppgift 2 (2 poäng): $N, M \leq 200$ och $Q \leq 1\,000$.
Deluppgift 3 (2 poäng): $N, M \leq 500$ och $Q \leq 200\,000$.
Deluppgift 4 (4 poäng): $N \cdot M \leq 1\,000\,000$ och $Q \leq 2\,000\,000$.

Exempel. $N = 3, M = 4, Q = 2$ och det ursprungliga rutnätet är

3	−2	5	0
7	2	−4	1
4	9	−3	−5

Den första instruktion ökar talet i rad 3 och kolumn 3 från −3 till 6, och det resulterande rutnätet blir då

3	-2	5	0
7	2	-4	1
4	9	6	-5

Talen 7 och 9 är störst i både sina rader och i sina kolumner, och det finns inga andra tal som är störst i både sin rad och sin kolumn, så svaret som ska skrivas ut efter den första instruktionen är 2.
Den andra instruktionen ökar talet i rad 1 och kolumn 4 från 0 till 8, och det resulterande rutnätet blir då

3	-2	5	8
7	2	-4	1
4	9	6	-5

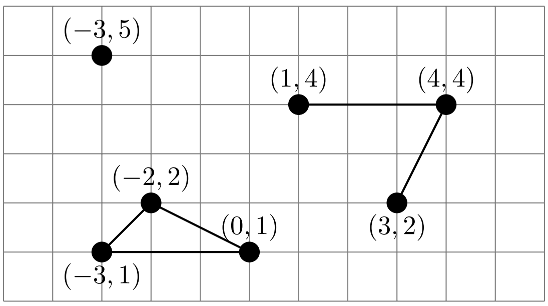
Det finns nu tre tal som är störst i både sina rader och i sina kolumner, så svaret som ska skrivas ut efter den andra instruktionen är 3.

Problem 2. Givet är N punkter $V = (x_i, y_i)$ (där $1 \leq i \leq N$). Betrakta nu grafen med V som hörn och en kant mellan (x_i, y_i) och (x_j, y_j) om och endast om $|x_i - x_j| + |y_i - y_j| \leq M$, där M är en given konstant. Samtliga koordinater uppfyller $-10^9 \leq x_i, y_i \leq 10^9$.

Beskriv ett program som beräknar antalet sammanhängande komponenter i denna graf.

Deluppgift 1 (1 poäng): $N \leq 1\,000$, $M \leq 10^9$.
Deluppgift 2 (1 poäng): $N \leq 100\,000, M \leq 50$.
Deluppgift 3 (3 poäng): $N \leq 100\,000, M \leq 1\,000$.
Deluppgift 4 (5 poäng): $N \leq 10^7$, $M \leq 10^9$.

Exempel. $N = 7, M = 3$ och punkterna är placerade som följer:



Kanterna i den resulterande grafen har också ritats ut, och det framgår av figuren att det finns 3 sammanhängande komponenter i grafen.

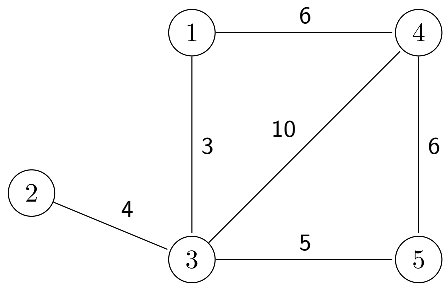
Problem 3. Givet är en viktad, oriktad graf med N hörn och M kanter. För en sekvens av hörn $X = (x_1, \ldots, x_k)$, låt $f(X)$ vara längden på den kortaste stigen som besöker varje hörn i sekvensen i ordning utan att använda samma kant två gånger i rad, alternativt ∞ i de fall en sådan stig inte finns.

Givet är dessutom en sådan sekvens X av längd K samt Q uppdateringar, där varje uppdatering ändrar ett givet värde i sekvensen till ett annat. Observera att längden på sekvensen alltid är oförändrad.

Beskriv ett program som efter varje uppdatering beräknar $f(X)$.

Deluppgift 1 (1 poäng): $Q, N, M, K \leq 10$.
Deluppgift 2 (4 poäng): $N, M \leq 1\,000$, $Q \leq 100\,000$, $K \leq 10\,000$.
Deluppgift 3 (5 poäng): $N, M \leq 1\,000$, $Q \leq 100\,000$, $K \leq 100\,000$.

Exempel. $N = 5, M = 6, K = 3, Q = 2$ och grafen ser ut som följer:



Det ursprungliga värdet på X är $(4, 3, 4)$ och den första uppdateringen ändrar X till $(4, 2, 4)$. Svaret som ska skrivas ut efter den första uppdateringen är då ∞, för efter att ha kommit till nod 2 finns det inget sätt att ta sig vidare utan att använda samma kant två gånger i rad.
Den andra uppdateringen ändrar X till $(4, 2, 2)$ och svaret som ska skrivas ut efter denna uppdatering är 13.

Problem 4. Givet är $3N$ positiva heltal $0 \leq a_1, a_2, \ldots, a_{3N} \leq 10^9$ och ett positivt heltal $K \leq N$. Vi säger att en mängd $I \subseteq \{1, 2, \ldots, 3N\}$ är K-*begränsad* om det gäller för varje $i \in \{1, 2, \ldots, 2N + 1\}$ att I innehåller högst K av talen $i, i + 1, \ldots, i + N - 1$.

Beskriv en algoritm som beräknar det största värdet som summan $\sum_{i \in I} a_i$ kan anta då I är en K-begränsad mängd.

Deluppgift 1 (1 poäng): $N \leq 100$ och $K = 1$.
Deluppgift 2 (1 poäng): $N \leq 100$ och $K \leq 3$.
Deluppgift 3 (5 poäng): $N \leq 200$ och $K \leq 10$.
Deluppgift 4 (3 poäng): $N \leq 500$ och $K \leq 20$.

Exempel. $N = 2, K = 1$ och

$$(a_1, a_2, a_3, a_4, a_5, a_6) = (4, 5, 3, 1, 3, 5).$$

I detta fall maximeras summan då $I = \{1, 3, 6\}$, och svaret blir därför $a_1 + a_3 + a_6 = 12$.
Notera att mängden $I = \{1, 2, 6\}$ ger en större summa, men den är inte K-begränsad eftersom den innehåller både 1 och 2.