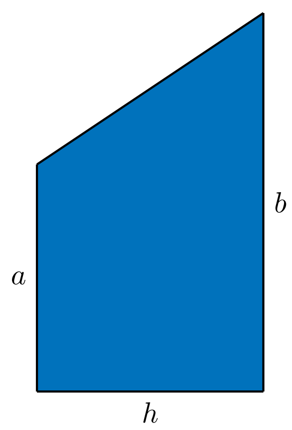
Soluții pentru problema *Trapez*

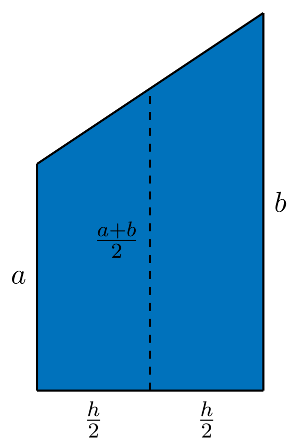
Lotul Național de Informatică, Craiova 2025

În această problemă, se dă o mulțime de puncte cu coordonate pozitive, și ni se cere, dându-se două laturi $x = x_0, x = x_1$ paralele cu axa Oy, să găsim trapezul de arie minimă delimitată de $x = x_0, x = x_1$ și de axa Ox, care să conțină toate punctele cu $x_0 \leq x \leq x_1$.

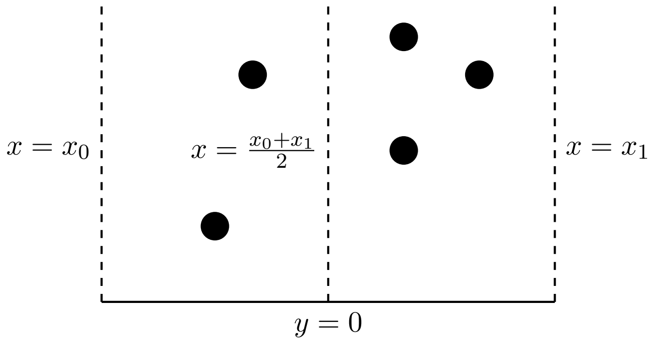
Mai întâi să determinăm analitic care este trapezul optim. Să considerăm trapezul de mai jos.



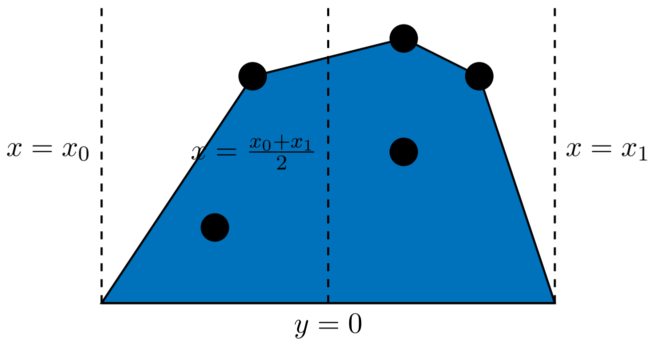
Aria lui este $\frac{a+b}{2}h$. Echivalent, observăm că $\frac{a+b}{2}$ este dat de lungimea înălțimii din următorul grafic.



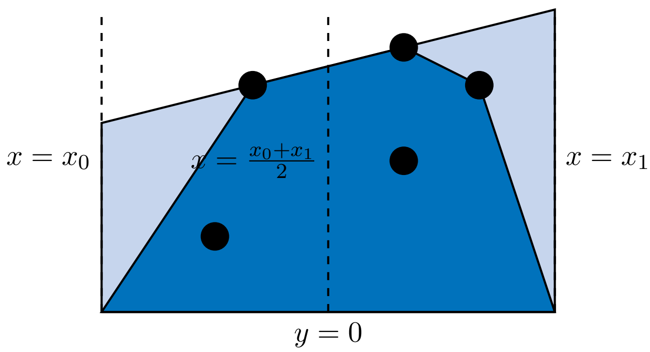
Prin urmare, putem repeta problema altfel: date fiind o mulțime de puncte cu coordonatele între x_0 și x_1, să se găsească y minim astfel încât să existe o dreaptă ce trece prin $(\frac{x_0+x_1}{2}, y)$ care să aibă sub ea toate punctele din mulțime. Să considerăm următorul exemplu, cu $x_0 = 0, x_1 = 6$.



Dorim practic dreapta care este peste toate punctele date, care să intersecteze dreapta $x = \frac{x_0+x_1}{2}$ cât mai jos. Observăm că dreptele care intersecteaza o dreaptă de forma $x = a$ cât mai jos sunt exact dreptele care prelungesc segmentele din înfășurătoarea convexă a punctelor. Construim mai întâi înfășurătoarea:



Apoi, prelungim segmentul ce intesectează $x = \frac{x_0+x_1}{2}$ pentru a găsi trapezul dorit.



Așadar, este suficient să putem găsi infășurătoarea convexă a punctelor din subsecvențele de interogare, într-o formă prin care putem interoga punctele cele mai aproape (ca coordonate x) de $\frac{x_0+x_1}{2}$. Acest lucru se poate realiza în complexitate $O((N+Q)\log N)$, dar vom descrie o soluție mai simplă, în timp $O((N+Q)\log^2 N)$, care putea lua 100 de puncte.

Soluția se bazează pe combinarea a două idei. În primul rând, putem menține înfășurătoarea convexă a punctelor folosind treap-uri persistente. Operația de bază pe care o vrem este una de tip *merge*: date fiind două înfăsurători convexe, una cu toate coordonatele x strict mai mici ca cele din cealaltă, să se genereze înfășurătoarea convexa a reuniunii punctelor. Dacă menținem înfășurătorile cu treap-uri, atunci putem să căutăm binar in fiecare treap care este care este ultimul punct care va fi inclus in înfășurătoarea rezultat — iar pentru a verifica dacă un punct oarecare va fi inclus, folosim din nou căutare binară. Așadar complexitatea unei operații de acest tip va fi $O(\log^2 n)$. Atunci când una dintre înfășurători constă din doar un punct, această operație se poate optimiza la $O(\log n)$.

A doua idee este cea a tehnicii divide et impera. Vom împărți punctele date în funcție de o coordonata x_m cât mai egal. Vom răspunde apoi la toate interogările ce conțin coordonata x_m. Pentru a face asta, vom calcula pentru fiecare prefix/sufix în jurul lui x_m înfășurătoarea convexă, folosindu-ne de faptul că folosim treap-uri *persistente* pentru a le ține pe toate în memorie simultan. Apoi, putem să calculăm înășurătoarea pentru un segment $[x_0, x_1]$ dând un merge între un prefix și un sufix. Per total, complexitatea va fi de $O((n+q)\log^2 n)$.