David
Pozdravljeni, moje ime je David, z menoj sta Marcel in Mateja. Danes vam bomo predstavili hibridni algoritem d22 z uporabo požrešne metode in vračanjem kot rešitev logističnega problema skupnega potovanja.

Marcel
Imamo naslednji problem: za delo v skladišču smo kupili nove drone. Zamislili smo si naslednji test letenja: dva drona bomo postavili nekam v skladišče in jima določili končne koordinate. Nato bomo opazovali njun let do teh koordinat. Program mora izpisati pot dronov, preprečiti moramo trk (drona nikoli ne smeta biti na istih koordinatah).

Pri tem pa imamo dve omejitvi: zaporedni poziciji drona se lahko razlikujeta v največ eni koordinati in to za največ 1 (dron se torej lahko premakne za največ 1 v smeri x, y ali z; diagonalni premiki niso mogoči). Dron lahko vedno stoji na mestu. Druga omejitev nam določa, da se drona ob istem času nikoli ne smeta nahajati na isti poziciji.

Mateja
Obravnavani problem je eden večjih problemov logistike, na primer v avtomatiziranih sistemih za shranjevanje in iskanje z uporabo robotov ali dronov, kot na primer skladišče trgovca Amazon.

Nalogo prav tako lahko prevedemo na problem iskanja poti, na primer platforma Google Maps. Vojska je naslednji uporabnik rešitve tega problema, kjer je naloga veliko bolj podobna obravnavani tematiki - tudi tam se, na primer, v primeru obravnave leta letalov, želimo izogniti trčenju, pri čemer morata biti letali čim bližje zaradi varnosti in podpore v primeru težav.

David
Problem lahko matematično opišemo z uporabo teorije grafov in vektorji. Definirajmo V kot končno neprazno množico in E kot družino dvoelementnih podmnožic množice V. Urejenemu paru G=(V,E) pravimo graf na množici vozlišč V in z množico povezav E.

Če vozlišči *uv* tvorita povezavo, potem rečemo, da sta *u* in *v* sosednji vozlišči oziroma sosedi.
Če povezava {u, v} določa smer, torej je {u, v} != {v, u} imamo usmerjen graf, v nasprotnem primeru je graf neusmerjen. Če povezave vsebujejo uteži, ki predstavljajo razdaljo med vozliščema, imamo utežen graf. V nasprotnem primeru je graf neutežen. Na sliki 1 lahko vidimo neutežen, neusmerjen graf G.

Vektor je urejen par točk v prostoru, ki je določen z velikostjo in smerjo. Nastopa lahko v katerem koli koordinatnem sistemu, za naš problem bomo uporabili tridimenzionalni pravokotni kartezični sistem, ki je določen z tremi osmi – os x, y in z.

Na sliki 2 lahko vidimo vektor AB, na sliki 3 pa primer tridimenzionalnega prostora z dvema točkama: A(2,2,0) in B(2,0,1) in njuna pripadajoča krajevna vektorja – krajevni vektor je daljica od koordinatnega izhodišča do točke T.

S spodnjo formulo pa izračunamo razdaljo med točkama v 3d prostoru.

Mateja

Teorijo grafov in vektorje lahko uporabimo za sestavo rešitve podanega problema. Položaj drona v 3d prostoru lahko predstavimo kot vozlišča, povezave med vozlišči pa kot vektorje.
Primer takšnega grafa je viden na sliki 4. Pričetek drona je v p0, ki ima koordinate (0,0,0), konec pa v p7, ki ima koordinate (1,1,1). Z rjavo barvo je označena ena od rešitev.

Marcel

Algoritem z grobim pristopom je sestavljen iz treh delov. Najprej generiramo vsa vozlišča, nato vse povezave, na koncu pa izvedemo grobi pristop. Generiranje vrednosti za koordinato prikazuje algoritem 1, generiranje vozlišč algoritem 2, ustvarjanje povezav pa algoritem 3.

Na koncu izvedemo algoritem grobe sile – jemljemo vozlišča, dokler nismo na cilju. Če je vozlišče veljavno in ne pride do trka, sprejmemo vozlišče. V nasprotnem primeru izvedemo mirovanje ali se vrnemo nazaj.

Mateja

Pri algoritmu d22 namesto, da ustvarimo vsa vozlišča, jih sedaj ustvarjamo po potrebi - algoritma 5 in 6. Sam algoritem d22 je zelo podoben metodi grobega pristopa, razlikuje se samo v dinamičnosti ustvarjanja vozlišč oziroma povezav.

David

Če generiramo vse povezave grafa, dobimo n * (n + 1) / 2 povezav oziroma polni graf, kar predstavlja kvadratno časovno zahtevnost. Če povezave ustvarjamo dinamično, dobimo 3*n / 2 povezav, kar predstavlja linearno časovno zahtevnost.

Slika 5 prikazuje polni graf, pri čemer je začetna točka v (0,0,0), končna točka pa v (2,4,4,). Kako hitro narašča število povezav pri polnem grafu v primerjavi z upoštevanjem omejitve problema, prikazuje slika 6.

Definirajmo začetne koordinate drona A kot prvo vrstico matrike A, njegove končne koordinate pa kot drugo vrstico matrike A. Enako storimo za dron B. Za dron A bomo z metodo grobe sile ustvarili 45.756 vozlišč in kar 1 miljardo povezav. Za dron B bomo ustvarili 6.624 vozlišč in 22 miljonov povezav.

Če uporabimo formulo za izračun razdalje med točkama, ugotovimo, da bomo potrebovali 62 oziroma 32 vozlišč. Primer polnega grafa z petdesetimi vozlišči je viden na sliki 7, v našem primeru pa imamo 45k oziroma 6k vozlišč.

Z algoritmom d22 bomo ustvarili 108 in 57 vozlišč in 162 in 85 povezav, kar je veliko manj kot prej.

Sedaj pa bomo prikazali delovanje programa.

(zagon programa)

Marcel

Pri prejšnjem primeru se je algoritem grobe sile izvajal 1.3 sekunde, porabil pa 522 MB pomnilnika. Algoritem d22 se je izvedel v 0.009 sekunde, porabil pa 5.6 MB pomnilnika.

Mateja

Pri tem grafu je opazna razlika v času izvajanja. Pri enostavnih primerih sta algoritma skoraj izenačena, pri težjih primerih pa pride do očitne razlike.

Zgodba je podobna pri porabi pomnilnika. Pri enostavnih primerih je poraba podobna, pri težjih primerih pa algoritem d22 porabi veliko manj pomnilnika – pri zadnjem primeru iz 522 MB pri grobem pristopu na 6 MB pri algoritmu d22.

David

Metoda grobe sile ima časovno zahtevnost O(n^2) (kvadratna zahtevnost), medtem ko ima d22 časovno zahtevnost O(n) (linearna zahtevnost).

Metoda grobe sile ima prostorsko zahtevnost S(n^2) (kvadratna zahtevnost), medtem ko ima d22 prostorsko zahtevnost S(1) (konstantna zahtevnost).

Razliko v časovni zahtevnosti prikazuje slika 10, medtem ko razliko v prostorski zahtevnosti prikazuje slika 11.