

پیدا کردن نزدیکترین نقطه و فواصل آنها در یک گراف با استفاده از الگوریتم تقسیم و حل

این روش را می توان به سه قسمت زیر تقسیم کرد:

تقسیم: این شامل تقسیم مسئله به برخی از زیر مسئله ها است.

حل: زیر مسئله با فراخوانی بازگشتی تا حل نقاط فرعی.

ترکیب مسئله حل شده تا بتوانیم راه حل مسئله را پیدا کنیم.

از یک آرایه از n نقطه در صفحه استفاده کردیم . مسئله این هست که نزدیکترین جفت نقاط و فواصل نقاط را در آرایه پیدا کنیم. به

عنوان مثال ، در کنترل ترافیک هوایی ، ممکن است بخواهیم هواپیماهایی را که بیش از حد به هم نزدیک هستند نظارت کنیم ، که

این ممکن است بر اثر نشان دهنده برخورد احتمالی آنها باشد.

ورودی: آرایه ای از n نقطه P

خروجی: کمترین فاصله بین دو نقطه در آرایه ی داده شده.

(نقطه میانی را در آرایه مرتب شده پیدا می کنیم، تا $P[n/2]$ را به عنوان نقطه میانی در نظر بگیریم.

(آرایه داده شده را به دو نیمه تقسیم می کنیم. زیر مجموعه اول شامل نقاطی از $P[0]$ تا $P[n/2]$ است.

چون آرایه دوم شامل نقاطی

از $P[n/2 + 1]$ تا $P[n-1]$ است.

به صورت بازگشتی کمترین فاصله را در هر دو زیر آرایه پیدا می کنیم. اگر فاصله ها d_l و d_r باشد. یعنی

راست و چپ) حداقل d_l

و d_r را پیدا می کنیم. تا حداقل d باشد.

از 3 مرحله قبل ، یک حد بالایی حداقل فاصله داریم. حال باید جفت نقاط رو را به این صورت در نظر

بگیریم که یک نقطه از جفت

از نیمه چپ و دیگری از نیمه راست باشد. یعنی نقاط عبوری از $P[n/2]$ و تمام نقاطی را که مختصات

x آنها نزدیکتر از d به نقاط

دیگر هست پیدا می کنیم. از همه این نقاط یک آرایه می سازیم.

آرایه را با توجه به مختصات y مرتب می‌شود. این مرحله $O(n \log n)$ است. با مرتب سازی مجدد و ادغام می‌توان آن را $O(n)$ به بهینه کرد.

کمترین فاصله را در آرایه یا گراف پیدا می‌کنیم ، یک مرحله $O(n^2)$ هست اگر نامرتب باشد، اما در واقع $O(n)$ است.

در آخر حداقل d و فاصله محاسبه شده در مرحله بالا را در مرحله 6 دوباره انجام می‌دهیم. (مرحله 6 پیچیدگی زمان: پیچیدگی زمان الگوریتم اگر $T(n)$ باشد. فرض می‌کنیم که اگر از الگوریتم مرتب سازی $O(n \log n)$ استفاده می‌کنیم.

کنیم. الگوریتم فوق همه نقاط را در دو مجموعه تقسیم می‌کند و به صورت بازگشتی دو مجموعه را فرا می‌خواند. پس از تقسیم ،

نقاط را در زمان $O(n)$ پیدا می‌کند ، نقاط را در زمان $O(n \log n)$ مرتب می‌کند و در نهایت نزدیک ترین نقاط را در گراف در

زمان $O(n)$ پیدا می‌کند. بنابراین $T(n)$ می‌تواند به صورت زیر باشد:

$$T(n) = 2T(n/2) + O(n) + O(n \log n) + O(n)$$

$$T(n) = 2T(n/2) + O(n \log n)$$

$$T(n) = T(n \log n * \log n).$$

