# جزوه جلسه بیست و شش داده ساختارها و الگوریتم

۳۰ آذر ۱۴۰۰

## فهرست مطالب


۱ مسئله پرسش کمینه یک محدوده ۲
۱.۱ راه حل ساده . . . . . . . . . . . . . . . . . . . . . . . . . ۲
۲.۱ ایده جذر . . . . . . . . . . . . . . . . . . . . . . . . . ۲
۳.۱ ایده توان های ۲ . . . . . . . . . . . . . . . . . . . . . . . . ۳
۴.۱ درخت پاره ای . . . . . . . . . . . . . . . . . . . . . . . . ۳
۱.۴.۱ درخت دکارتی . . . . . . . . . . . . . . . . . . . . . . ۴

# ۱ مسئله پرسش کمینه یک محدوده

یک لیست به طول $n$ داریم. هر بار عضو کمینه یک بازه از لیست مورد پرسش قرار می گیرد. راه حل های مختلف پوشش یافته در این بخش عبارتند از:

۱. راه حل ساده!

۲. ایده جذر

۳. ایده توان های ۲

۴. درخت پاره ای

۵. راه حل بهینه

## ۱.۱ راه حل ساده

در راه حل ساده، در هر Query، با یک حلقه از ابتدا تا انتهای بازه و بدون انجام هیچ پیش پردازشی، مقدار مینیمم محاسبه می شود.
پس زمان های زیر را داریم:

- پیش پردازش: $\Theta(1)$
- حافظه اضافی: $\Theta(1)$
- پرسش: $\Theta(n)$
- تغییر عنصر: $\Theta(1)$

## ۲.۱ ایده جذر

در این راه حل، آرایه را به قسمت هایی به طول $\sqrt{n}$ تقسیم می کنیم و مقدار minimum را برای هر کدام ذخیره می کنیم. برای پاسخ به هر پرسش نیز با فرض بازه [i, j]، بازه هایی که خود i و j در آن حضور دارند و تمام بازه هایی بین آن ها را بررسی می کنیم و مقدار مینیمم را از بین مینیمم های کاندید انتخاب می کنیم. برای تغییر عنصر نیز، جایگاه آن را در بازه مربوط به خود پیدا میکنیم و عنصر را تغییر می دهیم و در صورت کوچک بودن مقدار جدید از مقدار مینمم، مینیمم را نیز به مقدار جدید تغییر می دهیم.
توجه شود که بازه به طول $\sqrt{n}$ بهینه ترین حالت ممکن برای طول های مختلف است و زمانی بهتر از $\Theta(\sqrt{n})$ نمی توان متصور شد. همچنین ایده استفاده شده در این مسئله بسیار کاربردی است برای زمان های زیر نیز داریم:

- پیش پردازش: $\Theta(n)$
- حافظه اضافی: $\Theta(\sqrt{n})$
- پرسش: $\Theta(\sqrt{n})$
- تغییر عنصر: $\Theta(1)$

## ۳.۱ ایده توان های ۲

برای هر j در بازه [1, n]، بازه هایی که از j شروع می شوند و طول $2^i$ دارند ($0 \leq i \leq log_2n$) را در نظر می گیریم و کمینه آن را در پیش پردازش به شکل زیر محاسبه می کنیم:
$min(A[j, j + 2^i - 1] = min(min(A[j, j + 2^{i-1} - 1]), min(A[j + 2^i, j + 2^i - 1 + 2^{i-1}]))$
با انجام پیش پردازش بالا، پاسخ به هر پرسش در زمان (1) به شکل زیر داده می شود:
$min(A[i, k] = min(min(A[j, j + 2^i - 1]), min(A[k - 2^i + 1, k]))$
توجه داشته باشید که برای رسیدن به پاسخ یک پرسش، می توان بازه را به چند زیر بازه تقسیم کرد و مینیمم آن ها را پیدا کرد در حالی که بازه ها باهم اشتراک داشته باشند. برای مثال اگر طول بازه مورد پرسش برابر با ۱۵ باشد، در نظر گرفتن دو بازه به طول هشت ما را به جواب می رساند. اما باید پیدا کردن طول بازه مد نظر برای هر طول بازه ای در $\Theta(1)$ انجام بگیرد. بدین منظور با انجام یک پیش پردازش دیگر در $\Theta(n)$، برای هر عدد بزرگترین توان ۲ کوچکتر مساوی آن را ذخیره می کنیم. رابطه بالا برای رسیدن به پاسخ هر Query نیز از این پیش پردازش استفاده می کند.
در نهایت زمان عملیات برای این راه حل برابر است با:

- پیش پردازش: $\Theta(nlogn)$
- حافظه اضافی: $\Theta(nlogn)$
- پرسش: $\Theta(1)$
- تغییر عنصر: در این راه حل، تغییر عنصر نداریم.

## ۴.۱ درخت پاره ای

برگ های درخت را برابر با اعداد آرایه در نظر می گیریم و هر راس درونی برابر است با مینیمم فرزندانش. حافظه اضافی در این راه حل از مرتبه $\theta(n)$ می باشد. پیش پردازش نیز از برگ ها شروع و به ریشه ختم می شود و زمان کل آن نیز برابر است با $\Theta(n)$.
پرسش ها نیز در زمان $\Theta(logn)$ (به طور دقیق در $2logn$) انجام می شود.

- پیش پردازش: $\Theta(n)$
- حافظه اضافی: $\Theta(n)$
- پرسش: $\Theta(logn)$
- تغییر عنصر: $\Theta(logn)$

### ۱.۴.۱ درخت دکارتی

داده ساختار درختی دیگر که می توان از آن استفاده کرد، درخت دکارتی است. مینیمم آرایه کل در ریشه، عناصر سمت چپ آن در زیر درخت چپ و عناصر سمت راس در زیر درخت چپ به صورت بازگشتی درخت را تشکیل می دهند.
با داشتن یک استک می توان درخت را ساخت. آرایه را از چپ می خوانیم و در استک می ریزیم. در هر لحظه اگر عنصر جدید کوچکتر از سر استک بود، عناصر موجود در استک را تا جایی که سر استک کوچک تر از عضو جدید باشد پاک می کنیم و به سمت چپ عنصر جدید اضافه می کنیم و عضو جدید را در استک اضافه می کنیم. اما اگر عضو جدید بزرگتر از سر استک بود، به استک اضافه شده و در سمت راست سر استک قرار می گیرد.
به دلیل نزدیکی این درخت و مسئله Range Minimum Query (RMQ) با داشتن RMQ نیز می توان درخت را ساخت.
همچنین می توان درخت را تشکیل داد و در $\Theta(1)$ به مسئله RMQ پاسخ داد. بدین منظور برای بازه [i, j] ، پایین ترین جد مشترک Lowest Common Ancestor (LCA) دو راس A[i] و A[j] را پیدا می کنیم و به عنوان پاسخ مسئله خروجی می دهیم. تنها چالش باقی مانده پیدا کردن پایین ترین جد مشترک در $\Theta(1)$ است که با انجام پیش پردازش هایی این امر نیز ممکن می شود.