

تشریحی

به سوالات زیر پاسخ دهید:

(1) مرتب‌سازی پایدار (Stable Sort) چیست؟

در چه شرایطی استفاده از الگوریتم‌های پایدار می‌تواند نسبت به الگوریتم‌های ناپایدار مفیدتر باشد؟

(2) مرتب‌سازی درجا (In-place Sort) چیست؟

در چه شرایطی استفاده از الگوریتم‌های درجا می‌تواند نسبت به الگوریتم‌های غیر درجا مفیدتر باشد؟

(3) مقایسه Merge Sort و Quick Sort

الگوریتم مرتب‌سازی سریع (Quick Sort) و مرتب‌سازی ادغامی (Merge Sort)، هر دو پیچیدگی زمانی از مرتبه $O(n \log n)$ دارند؛ چه بسا مرتب‌سازی سریع در بدترین حالت $O(n^2)$ است اما همچنان خیلی از افراد به‌جای مرتب‌سازی ادغامی از مرتب‌سازی سریع استفاده می‌کنند. چرا؟

(4) برای هر الگوریتم مرتب‌سازی نام‌برده شده، موارد زیر را بنویسید:

- مشخص کنید:
 - پایدار یا ناپایدار
 - درجا یا غیر درجا
 - مرتبه زمانی در حالت‌های بهترین، متوسط و بدترین
- برای ورودی داده‌شده، مراحل مرتب‌سازی **صعودی** را گام به گام برای الگوریتم‌های شماره ۱، ۲، ۳ و ۵ نمایش دهید. (نیازی به توضیح کامل نیست؛ صرفاً خلاصه‌ای از روند کار همراه با نمایش کامل مراحل مرتب‌سازی کافی است.)
- مزایا و معایب هر الگوریتم را بنویسید و توضیح دهید که در چه شرایطی استفاده از آن بهتر است.

الگوریتم‌ها:

۱. مرتب‌سازی حبابی (Bubble Sort)
۲. مرتب‌سازی درجی (Insertion Sort)
۳. مرتب‌سازی انتخابی (Selection Sort)
۴. مرتب‌سازی سریع (Quick Sort)
۵. مرتب‌سازی ادغامی (Merge Sort)
۶. مرتب‌سازی شمارشی (Counting Sort)
۷. مرتب‌سازی پایه‌ای (Radix Sort)

آرایه ورودی:

[32, 14, 18, 32, 14, 14]

پیاده سازی 1

ترکیب الگوریتم های مرتب سازی رویکردی است که به طور خاص در مواردی که یک الگوریتم برای همه انواع داده ها بهینه نیست، به کار می رود. برخی الگوریتم ها در بدترین حالت پیچیدگی زمانی بالایی دارند، درحالی که برخی دیگر ممکن است به حافظه زیادی نیاز داشته باشند. بنابراین، انتخاب پویا و ترکیب چندین روش می تواند عملکرد کلی را بهبود بخشد. در این تمرین، شما باید دو الگوریتم "**مرتب سازی سریع (Quick sort)**" و "**مرتب سازی پایه ای (Radix sort)**" را به صورت ترکیبی طوری پیاده سازی کنید که الگوریتم مناسب را بر اساس ویژگی های ورودی فراخوانی کند.

(اگر دو الگوریتم رو جداگانه پیاده سازی کنید هم نمره کامل را می گیرید، ترکیبی پیاده سازی کنید (مشخص کردن آستانه) نمره امتیازی می گیرید.)

پیاده سازی 2

ترکیب الگوریتم های مرتب سازی به گونه ای طراحی می شود که بتوان در شرایط مختلف ورودی از نقاط قوت هر رویکرد بهره برد و در عین حال نقاط ضعف آن ها را تا حد امکان کاهش داد. در این روش، هنگامی که شرایط ورودی اجازه استفاده از الگوریتمی با زمان اجرای خطی را می دهد، مانند مواقعی که داده ها در بازه ای محدود قرار دارند، می توان از روشی بهره برد که در این شرایط عملکرد بسیار سریعی ارائه می دهد؛ اما در مواقعی که دامنه ورودی بسیار وسیع است یا ساختار داده ها پیچیده تر است، الگوریتمی با رویکرد تقسیم و حل می تواند عملکرد بهینه ای داشته باشد. هدف از ترکیب این دو رویکرد، استفاده از سرعت و کارایی در مواقعی که شرایط اجازه می دهد و همچنین تضمین پایداری و توانایی پردازش داده های پیچیده در سایر شرایط است. در این تمرین، شما باید دو الگوریتم "**مرتب سازی ادغامی (Merge sort)**" و "**مرتب سازی شمارشی (Counting sort)**" را به صورت ترکیبی طوری پیاده سازی کنید که الگوریتم مناسب را بر اساس ویژگی های ورودی فراخوانی کند.

(اگر دو الگوریتم رو جداگانه پیاده سازی کنید هم نمره کامل را می گیرید، ترکیبی پیاده سازی کنید (مشخص کردن آستانه) نمره امتیازی می گیرید.)

چینش متین - تقارن کلاسیک



متین و نیما دو فروشنده خلاق و البته رقیب قدیمی در بازار شهر هستند. هر کدام فروشگاه خود را دارند و هر روز تلاش می‌کنند که مشتری‌های بیشتری را جذب کنند. اما این بار رقابت به اوج خودش رسیده است. قرار است در جشنواره بزرگ فروش، هر کدام ویتترین مغازه اش را با یک ردیف از محصولات خاص بچیند؛ محصولاتی که هم از نظر قیمت منظم هستند و هم از نظر زیبایی چشم نواز.

متین ایده ای خاص دارد: او می‌گوید: محصولاتم رو این طور می‌چینم که قیمت‌های زوج، که نماد نظم و تعادل

هستن، از بیشترین به کمترین در سمت چپ ویتترین قرار بگیرن، و قیمت‌های فرد، که نماینده تنوع و جذابیت هستن، از کمترین به بیشترین در سمت راست.

نیما نیش خندی شیطانی می زند و می‌گوید: جالبه، ولی ببینم واقعاً می تونی همچین نظمی رو بدون کمک گرفتن از فضای اضافه و فقط با یه مرتب‌سازی هوشمندانه پیاده کنی؟

شما یک آرایه از اعداد دارید. باید این آرایه را طوری مرتب سازی کنید که اعداد زوج در سمت چپ آرایه مرتب شده به صورت نزولی قرار بگیرند و اعداد فرد در سمت راست آرایه به صورت صعودی.

ورودی :

خط اول: تعداد اعداد آرایه (n)

خط دوم: اعداد آرایه

خروجی :

آرایه‌ای که اعداد زوج به صورت نزولی در سمت چپ و اعداد فرد به صورت صعودی در سمت راست قرار دارند.

الزامات:

1) برای مرتب سازی، حتما الگوریتم **مرتب سازی انتخابی (selection sort)** را پیاده سازی کنید و از آن استفاده کنید.

2) از هیچ آرایه کمکی یا فضای اضافی برای ذخیره داده‌ها استفاده نشود، الگوریتم شما باید درجا (in-place) باشد.

(مثال)

ورودی نمونه 1:

8

7 2 5 8 4 3 6 1

خروجی نمونه 1:

8 6 4 2 1 3 5 7

چینش نیما - هنر ترکیب





بعد از نمایش خیره کننده متین در جشنواره، حالا نوبت نیما هستش که خلاقیت خودش را به نمایش بگذارد. او که همیشه طرفدار نوآوری و سبک های خاص بوده، تصمیم گرفته تا برخلاف متین که به نظم خشک و جدی علاقه دارد، ویتترین مغازه اش را با ترکیب هوشمندانه ای از تنوع و تعادل در زیبایی و قیمت بچیند. نیما می گوید:

من می خوام محصولاتم رو به شکلی بچینم که حس تعادل و زیبایی در هم آمیخته باشه. یک قیمت زوج، بعد یک قیمت فرد، دوباره زوج، بعدی فرد... و همین طور تا آخر! همه چیز باید مرتب و صعودی باشه تا چشم مشتری رو نوازش بده، اما اگر از یه نوع بیشتر داشتیم، مشکلی نیست؛ بقیه رو بی نظم ته ویتترین می ذاریم، چون خاص بودن همیشه یه جور به هم ریختگی قشنگ هم داره! در همین لحظه، متین با نگاهی موشکافانه به چینش نیما خیره می شود و با لبخندی تمسخر آمیز می گوید:

همیشه هم نظم خشک جواب نمی ده، نه؟ ولی ببینم نیما، می تونی فقط با یک الگوریتم هوشمند این ترکیب هنرمندانه رو پیاده کنی؟!

شما یک آرایه از اعداد دارید. باید این آرایه را طوری مرتب سازی کنید که اعداد زوج و فرد، مرتب شده به صورت صعودی، یک در میان در کنار یک دیگر قرار بگیرند. اگر تعداد اعداد زوج یا فرد با یکدیگر برابر نبود، اعداد زوج یا فرد باقی مانده در انتهای آرایه قرار بگیرند. ترتیب اعداد باقی مانده ای که در انتهای آرایه قرار می گیرند هم مهم است.

ورودی:

خط اول: تعداد اعداد آرایه (n)

خط دوم: اعداد آرایه

خروجی:

آرایه ای که اعداد زوج و فرد یک در میان کنار هم قرار دارند و به صورت صعودی مرتب هستند.

الزامات:

1) برای مرتب سازی، حتما الگوریتم **مرتب سازی درجی (insertion sort)** را پیاده سازی کنید و از آن استفاده کنید.

(مثال)

ورودی نمونه 1:

8
7 2 5 8 4 3 6 1

خروجی نمونه 1:

1 2 3 4 5 6 7 8

ورودی نمونه 2:

8
7 13 5 8 4 3 -27 1

خروجی نمونه 2:

-27 4 1 8 3 5 7 13

در نمونه 2 همانطور که مشاهده می کنید ، تعداد اعداد فرد بیشتر از تعداد اعداد زوج است و بعد از اینکه با اعداد زوج یک در میان قرار گرفتند (8، 1، 4، -27)، اعداد فرد باقی مانده با ترتیب در انتهای آرایه قرار گرفته اند (13، 7، 5، 3).

مسیر مقرون به صرفه

به شما دنباله a شامل اعداد صحیح به طول $2n$ داده می شود شما باید این مجموعه را به n جفت دو تایی تبدیل کنید که نماینده x و y یک نقطه در صفحه است با فرض اینکه هزینه جابه جایی از نقطه به نقطه دیگر اختلاف مقادیر x به اضافه y اختلاف مقادیر y آن باشد سعی کنید مجموعه از نقاط درست کنید که پیمایش در آن کمترین هزینه را داشته باشد و سپس مقدار آن را چاپ کنید.

ورودی:

خط اول: یک عدد صحیح است که نماینده n است، تعداد نقطه های ورودی
خط دوم: شامل $2n$ عدد صحیح است، هر یک عدد از دنباله a هستند

خروجی:

حداقل هزینه پیمایش در کل نقاط ساخته شده

(مثال)

ورودی نمونه ۱:

2
15 1 10 5

خروجی نمونه ۱:

9

اگر نقاط $(10,1)$ و $(15,5)$ را بسازیم هزینه پیمایش ما برابر $|5-1| + |15-10| = 9$ است و در تمام مجموعه خود پیمایش را انجام داده ایم.

ورودی نمونه ۲

3

10 30 20 20 30 10

خروجی نمونه ۲

20

می توانید نقاط $(20,20)$ ، $(10,30)$ و $(10,30)$ را تشکیل دهید و دقیقاً به ترتیب از آنها بازدید کنید. سپس هزینه مسیر خواهد بود $20=0+0+10+10=|30-30|+|10-10|+|30-20|+|10-20|$.

بزرگترین عدد ممکن

تعدادی عدد در اختیار دارید. باید این اعداد رو طوری به یکدیگر بچسبونید که بزرگترین عدد ممکن با این اعداد رو درست کرده باشید.

ورودی:

خط اول: تعداد اعداد (n)

خط دوم: اعداد

خروجی:

بزرگترین عددی که می توان با چسبوندن اعداد داده شده تشکیل داد.

الزامات:

الگوریتم مرتب سازی مورد استفاده در این مسئله رو خودتون پیاده سازی کنید.

(مثال)

ورودی نمونه 1:

3
4 902 68

خروجی نمونه 1:

902684

پیشنهاد نیما

سؤال 1: تحلیل انتخاب پیوت در QuickSort فرض کنید در الگوریتم QuickSort پیوت همیشه به صورت تصادفی انتخاب می شود.

- **الف)** اثبات کنید امید ریاضی تعداد مقایسه های لازم برای مرتب سازی یک آرایه n عضوی با QuickSort، در حد $O(n \log n)$ است.
- **ب)** در صورتی که همیشه کوچک ترین یا بزرگ ترین عنصر را به عنوان پیوت انتخاب کنیم، پیچیدگی زمانی الگوریتم را دقیقاً به دست آورید.

سؤال 2: مدل سازی احتمالاتی در انتخاب پیوت در الگوریتم Randomized QuickSort، احتمال اینکه عنصر k ام آرایه (از کوچک ترین به بزرگ ترین) به عنوان پیوت انتخاب شود را بنویسید. سپس امید ریاضی طول مسیر بازگشتی (Recursion Tree) را بر اساس این احتمال محاسبه کنید.

سؤال 3: تحلیل رفتار بدترین حالت نشان دهید که در بدترین حالت (Pivot همواره کوچک ترین یا بزرگ ترین عنصر)، تعداد مقایسه های الگوریتم QuickSort برابر با $2n(n-1)$ است. به صورت استقرایی رابطه بازگشتی را نوشته و ثابت کنید.

پارسای چیترا (امتیازی)



پارسا یک گیمر فوق العاده حرفه ای هستش که در حال بازی کردن با بازی محبوبش هستش. بازی محبوب پارسا یک بازی سبک استراتژی هستش که در اون پارسا در نقش پادشاه سرزمین دیزآباد قراره به سمت سرزمین های دیگه بره و با جنگیدن با سرزمین ها ، اون ها رو فتح کنه.

پادشاه با گرگ سوار (فرماندهی گرگ های درنده) در ازای پرداخت پول قرار داد بسته که بعد از هربار فتح یک سرزمین توسط پادشاه و افرادش، گرگ سوار k تا سرزمین رو برای اون ها به تنهایی فتح کنه. در این حین بعضی از سرزمین ها با خودشون کریستال تاریکی دارند که موجب می شه که با ایجاد امواج صوتی زنده برای گرگ ها باعث می شه اون ها نتونند به اون سرزمین ها نزدیک بشن و گرگ سوار اون ها رو فتح کنه و در نتیجه پادشاه و افرادش باید خودشون دست به کار شن و با جنگیدن با اون سرزمین، اون رو فتح کنند.

در مپ بازی یک سری توپ خونه های پراکنده وجود دارند که هر کدام روی یک سرزمین مشخص نشانه گرفتند. پادشاه با استفاده از اون ها می تونه سرزمین هایی رو که نشانه گرفتند رو با مصرف انرژی کمتر نابود کنه اما گرگ سوار در صورتی که شاه از توپخانه استفاده کنه، سرزمینی رو برای شاه فتح نمی کنه. گرگ سوار فقط وقتی k

تا سرزمین رو برای شاه فتح می کنه که شاه و افرادش خودشون یک سرزمین رو بدون استفاده از توپ خانه فتح کنند.

فتح یا همون نابود کردن هر سرزمین انرژی از پادشاه و افرادش صرف می کنه که در پایان جنگ کلی (یا همون یک دست بازی) به میزان انرژی کل صرف شده؛ بازیکن (که همون پارسا هستش) باید صبر کنه تا پادشاه و افرادش استراحت کنند و برای جنگ بعدی (یا همون دست بعدی) آماده شن و بازیکن بتونه یک دست دیگه بازی کنه.

از اون جا که پارسا یک گیر حرفه ای هستش همیشه در بازی پیروز می شه و تمامی سرزمین ها رو با کمترین مصرف انرژی فتح می کنه. اما اون دیگه خسته شده و نمی خواد دیگه رو فکر کردن اینکه با چه استراتژی سرزمین ها رو با کمترین مصرف انرژی فتح کنه، وقت بذاره. از اونجا که پارسا همچنین یک برنامه نویس ماهر هستش، تصمیم می گیره که کد یک چیت برای بازی بنویسه که با دریافت اطلاعات یک دست بازی ، به پارسا می گه با چه استراتژی سرزمین ها رو فتح کنه و حداقل انرژی مورد نیاز برای فتح سرزمین ها رو به اون بگه... به پارسا در نوشتن این چیت کمک کنید.

ورودی:

در خط اول تعداد سرزمین های دشمن ، به تعداد n
در خط i ام (از 2 تا n) به ترتیب اول میزان انرژی مورد نیاز برای فتح سرزمین و سپس نشانگر کریستال تاریکی و در آخر شناسه سرزمین داده می شود.
در خط بعدی تعداد توپ خانه ها ، به تعداد m
در خط i ام (از $n+1$ تا m) به ترتیب اول میزان انرژی مورد نیاز برای استفاده از توپ خانه و سپس شناسه سرزمینی که توپ خانه به آن نشانه گرفته است.
در خط بعدی مقدار k که حداکثر تعداد سرزمین هایی است که گرگ سوار بعد از هر بار فتح توسط پادشاه و افرادش، می تواند فتح کند داده می شود.

خروجی:

حداقل انرژی مورد نیاز برای فتح تمامی سرزمین ها

الزامات:

کد شما باید شی گرا باشه ولی نیاز به سخت گیری آن نیست.

(مثال)

ورودی نمونه 1:

4
14 0 315
25 1 301
9 1 255
53 0 200
2
5 255
10 301
1

خروجی نمونه 1:

29

توضیح نمونه 1:

- 1) سرزمین 255 رو با توپخانه نابود می کنیم -> 5
 - 2) سرزمین 301 رو با توپخانه نابود می کنیم -> 10
 - 3) سرزمین 315 رو توسط شاه و به همراه اون سرزمین 200 رو توسط گرگ سوار فتح می کنیم -> 14 .
- => جمع انرژی های مصرف شده: 29

.

ورودی نمونه 2:

10
8 0 210

49 0 516
33 1 323
19 0 102
26 1 101
28 0 499
49 0 503
12 0 300
28 1 297
50 1 277
5
6 102
9 277
7 297
9 210
10 277
2

خروجی نمونه 2:

83

توضیح نمونه 2:

- 1) سرزمین 210 توسط شاه و به همراه اون سرزمین های 516 و 503 توسط گرگ سوار فتح می کنیم -> 8
- 2) سرزمین 277 رو با توپخانه نابود می کنیم -> 9
- 3) سرزمین 101 توسط شاه و به همراه اون سرزمین های 102 و 499 توسط گرگ سوار فتح می کنیم -> 26
- 4) سرزمین 297 رو با توپخانه نابود می کنیم -> 7
- 5) فتح سرزمین 323 توسط شاه و افرادش . به همراه اون سرزمین 300 توسط گرگ سوار فتح می شه سرزمین بدون کریستال دومی وجود نداره برای گرگ سوار -> 33

. => جمع انرژی های مصرف شده: 83