بسته‌بندی بهینه

Team Name: FOUR

سید محمد هاشمی – احمد محمودیان درویشانی – محسن دهباشی – غزال ربیعی

فهرست

سیستم‌عامل موردنیاز

کد راه حل ارائه شده برای مسئله در سیستم‌عامل‌های مختلف قابل اجراست اما کارایی سیستم‌عامل لینوکس برای اجرای بهینه‌تر کدها قابل ذکر است. توزیع‌های مختلف سیستم‌عامل لینوکس از جمله Ubuntu 20.04 می‌تواند مورد استفاده قرار گیرد.

تکنولوژی و زبان مورداستفاده و ورژن آن

زبان مورداستفاده ما زبان C++ و ورژن C++11 آن می‌باشد که البته کد ارائه شده قابل اجرا با ورژن‌های دیگر بخصوص ورژن‌های بعدی این زبان نیز می‌باشد.

کتابخانه‌های مورداستفاده

در پیاده‌سازی راه حل این مسئله تنها از کتابخانه iostream زبان C++ استفاده شده است.

دیتابیس‌های موردنیاز و ورژن آن

در حل این مسئله از دیتابیس خاصی استفاده نشده است.

تشریح دقیق و روشن مراحل حل مسئله

درک و تحلیل مسئله:

ابتدا مسئله به طور دقیق خوانده شده خواسته آن برای اعضای تیم مشخص شد و در نهایت با تبیین تعدادی مثال به فضای مسئله مسلط شدیم.

تفکر و ایده‌پردازی:

در فضای مسئله با توجه به تحلیل‌های انجام شده به نتایجی دست یافتیم از جمله این که در صورت انتخاب j به عنوان اندیس یک شیء درصورت موفق بودن الگوریتم و بسته‌بندی موفق اشیاء j تا n قطعا میتوانیم اشیاء j+1 تا n را نیز با الگوریتم داده شده در مسئله با موفقیت بسته‌بندی کنیم و همچنین برای این کار تعداد بسته‌های لازم با انتخاب j+1 از تعداد بسته‌های لازم با انتخاب j کوچکتر مساوی است. پس موفقیت الگوریتم که وابسته به تعداد بسته‌های لازم برای بسته‌بندی است را با توجه به نزولی بودن تعداد بسته های مورد نیاز به ازای افزایش مولفه j می‌توان به یک مسئله Binary Search The Answer مدل کرد.

بیان ایده:

در این مسئله برای یافتن بیشینه تعداد اشیاء ممکن برای بسته‌بندی با الگوریتم گفته شده باید به دنبال کمینه اندیس باشیم که با در نظر گرفتن بازه دامنه که در اینجا [0, n-1] یعنی بازه انتخاب j است و بررسی امکان حل یا عدم امکان حل با انتخاب میانه بازه یعنی عضو mid = (n-1)/2 تصمیم می‌گیریم که حل مسئله را با

یکی از بازه‌های [0, mid] یا [mid+1, n-1] ادامه دهیم. این فرآیند به طور متوالی و حداکثر در اردر زمانی log n ادامه خواهد یافت و در نهایت به اندیس مطلوب برای j خواهیم رسید.

نوشتن شبه کد حل مسئله و تشریح آن

```
Input: n, m, k, array a[]
lo = 0
hi = n-1
While lo < hi:
     mid = (lo + hi) / 2

     // start doing the given algorithm in problem
     cnt = 1
     cap = 0
     for i in mid..n-1:
             if cap + a[i] <= k:
                     cap += a[i]
             else:
                     cap = a[i]
                     cnt += 1
     // end of algorithm given in problem
     if cnt <= m:
             hi = mid
     else:
             lo = mid + 1
Output: n - lo
```

متغیرهای lo و hi برای تعیین بازه انتخاب j در هر مرحله بکار می‌روند در هر مرحله میانه انتخاب و الگوریتم داده شده در صورت مسئله با آن اجرا می‌شود (متغیر cap برای محاسبه ظرفیت جعبه در حال پر شدن و متغیر cnt برای نگهداری تعداد جعبه‌های مورد نیاز ابتدا با ۰ و ۱ مقداردهی می‌شوند). تعداد جعبه‌های لازم برای بسته‌بندی اشیاء mid تا n-1 در متغیر cnt محاسبه شده است. حال اگر cnt از m که تعداد جعبه‌هاست کوچکتر مساوی باشد پس موفق بوده ایم و می‌توانیم به سراغ jهای کوچکتر برویم پس برای کوچک‌تر کردن بازه به سمت jهای کوچکتر hi = mid قرار میگیرد. اگرcnt از m بزرگتر باشد الگوریتم ناموفق بوده و باید به سراغ jهای بزرگتر برویم پس بازه را با lo = mid + 1 به سمت jهای بزرگتر کوچک می‌کنیم.

در نهایت عملیات زمانی که بازه به یک اندیس مشخص رسید (وقتی lo با hi برابر است) خاتمه می‌یابد و lo اندیس مطلوب برای j است و بیشینه تعداد اشیاء قابل بسته‌بندی با m جعبه برابر با n-lo است.

در انتهای این عملیات که با اردر زمانی O(log n) با نصف کردن بازه و انتخاب j طول می‌کشد و O(n) برای پردازش الگوریتم داده شده در مسئله در نهایت به اردر زمانی O(nlogn) دست می‌یابیم.

اردر حافظه این راه حل O(n) می‌باشد.

پیاده‌سازی شبه کد در غالب کد

شبه کد مرحله قبلی در قالب یک کد به زبان C++ نوشته شده و به زبان انگلیسی کامنت‌های کاملی برای تمام مراحل آن در نظر گرفته شده است. کد مورد نظر را از طریق لینک قرار داده شده در قسمت کد منبع در ادامه همین سند می‌توانید مشاهده نمایید.

کامپایلر موردنیاز و ورژن آن و سایر موارد لازم برای اجرای کد

برای کامپایل کد منبع می‌توانید از کامپایلر GNU G++ 11 5.1.0 استفاده کنید.

برای اجرا کد منبع بر روی سیستم‌عامل ویندوز می‌توانید دستورات زیر را به ترتیب وارد کنید:

- دستور کامپایل کد منبع: g++ main.cpp
- دستور اجرای کد منبع: a.exe < input.txt > output.txt

دستور فوق ورودی را از فایل input.txt خوانده به کد اجرا شده می‌دهد و خروجی را در فایل output.txt ذخیره می‌کند.

برای اجرا کد منبع بر روی سیستم‌عامل لینوکس می‌توانید دستورات زیر را به ترتیب وارد کنید:

- دستور کامپایل کد منبع: g++ main.cpp
- دستور اجرای کد منبع: ./a.out < input.txt > output.txt

دستور فوق ورودی را از فایل input.txt خوانده به کد اجرا شده می‌دهد و خروجی را در فایل output.txt ذخیره می‌کند.

کد منبع

لینک کد منبع در GitHub: https://github.com/smh997/Fanavard6/blob/Code/main.cpp