

به نام خدا

۵:۳۰

شروع کلاس

C++

زبان های برنامه نویسی

Python

مثال: n تا فایل که فایل نام طول L دارد.

برای دسترسی به فایل n ام باید $\sum_{k=1}^n L[k]$

هزینه کنیم.

مدت چیدن این فایل کمتر هم است به گونه ای

که کل زمان دسترسی به فایل ها کمینه شود.

تابع هزینه: $\sum_{k=1}^n \sum_{i=1}^k L[k]$

$\sum_{i=1}^n T[i]$

* اثبات: غرض کنیم k ترتیب فایل‌های باشد که الگوریتم در پیاده می‌دهد.

ک ترتیب

5	2	1	3	4
---	---	---	---	---

$5 + 7 + 8 + 11 + 15$

الگوریتم: بداس طول فایل‌ها ($T[i]$ ها) فایل‌ها را به ترتیب صعودی مرتب کنیم و بداس این ترتیب کنار هم بچینیم.

پیچیدگی الگوریتم: پیچیدگی الگوریتم مرتب‌سازی

$O(n \log n)$

به نام خدا

شروع کلاس ۵:۳۰

زبان های برنامه نویسی
C++
Python

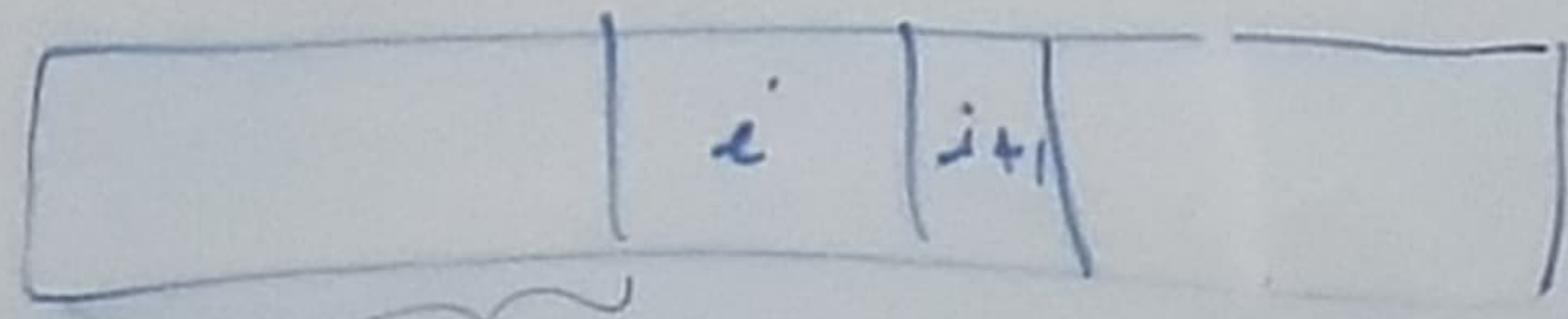
مدرس کنیم که یک ترتیب بهینه برای مسئله باشد

برهان خلف اگر تمام فایده ها به ترتیب صعودی در S قرار

گرفته باشند پس حداقل یک عنصر وجود دارد که
در ترتیب S

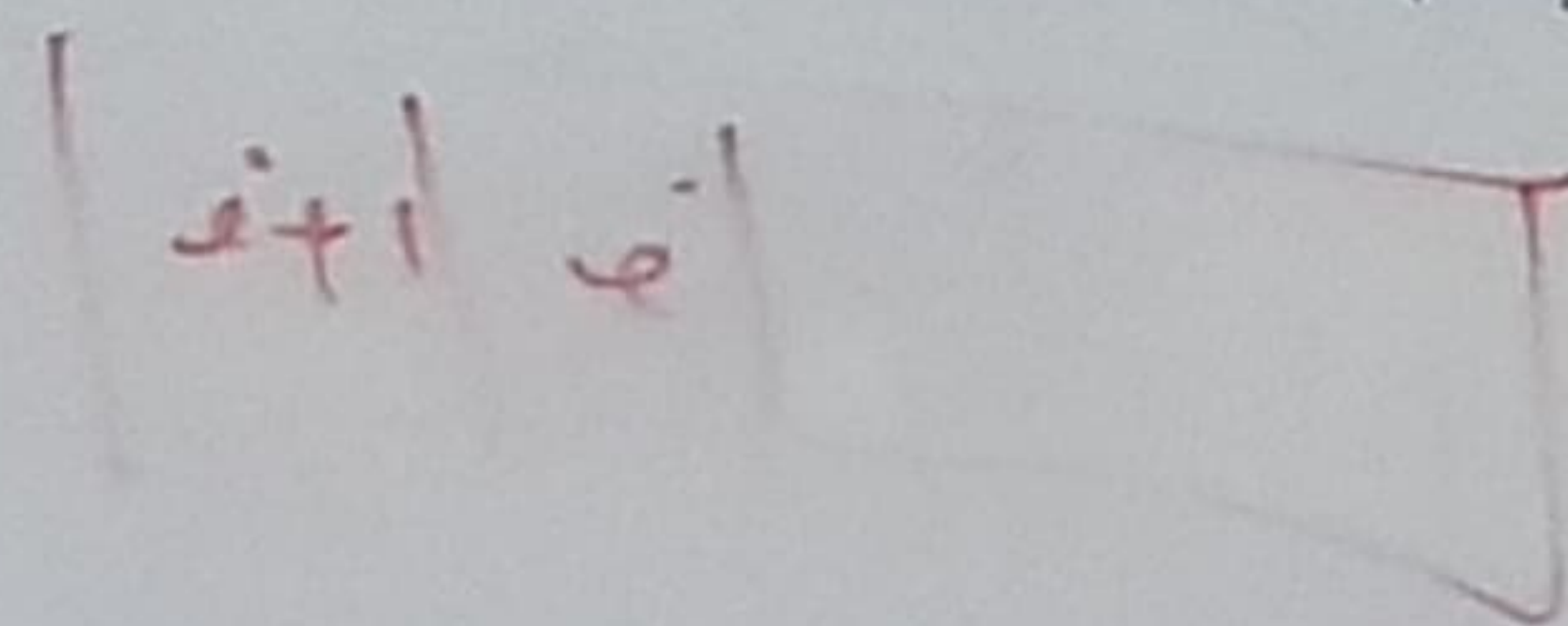
$[a, b], [c, d]$

S



زمان کل دسترس به ترتیب S :

S



حال با جابجایی در فایل‌ها و اند به یک ترتیب

می‌رسیم

$$T[s''] = T_1' + T_1 + L[i+1] + T_1' + L[i+1] + L[i]$$

$$T[s'] - T[s''] = L[i] - L[i+1] > 0$$

$$T[s''] < T[s']$$

که با بهینه بودن می‌در تناقض است.

$$T_1 = \sum_{k=1}^{i-1} t[k]$$

$$T_2 = \sum_{k=i+r}^n t[k]$$

$$T[k] = T_1' + T_1 + L[i] + T_1' + L[i] + L[i+1]$$

حزینة ترتیب ک

به نام خدا

شروع کلاس ۵:۴۰

زبان های برنامه نویسی
C++
Python

مثله زمان بندی فعالیت ها:

n تا فعالیت

که هر فعالیت f_i : زمان شروع : s_i

زمان پایان : t_i

هدف انتخاب بهترین تعداد فعالیت است به طوری که
هیچ دو فعالیت انتخاب شده همیشه نداشته باشند.

به نام خدا

۵:۴۰

شروع کلاس

C++

زبان های برنامه نویسی

Python

مسئله زمان بندی فعالیت ها:

n تا فعالیت

که هر فعالیت f_i : زمان شروع : s_i

زمان پایان : f_i

هدف : انتخاب بهترین تعداد فعالیت است به طوری که

هیچ دو فعالیت انتخاب شده همپوشانی نداشته باشند.

f_i	۱	۲	۳	۴	۵	۶
s_i	۱	۳	۵	۵	۴	۵
f_i	۴	۵	۶	۷	۹	۹

ابتدا مغالیت هارا به حسب زمان پایان به ترتیب

صعودی مرتب می کنیم. بدون کاستن از کلیت مطلب

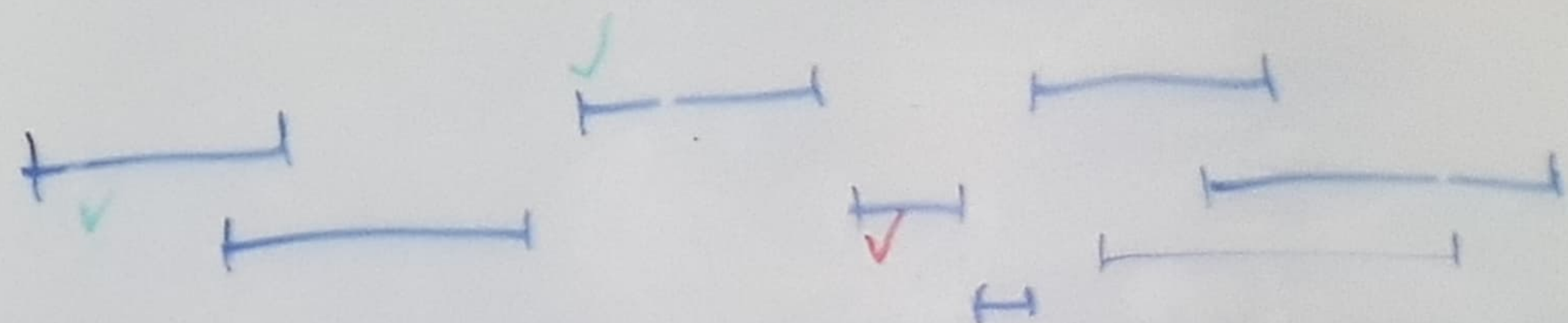
فرض می کنیم f_1, f_2, \dots, f_n

انتخاب حریصانه: در هر گام مغالیتی انتخاب شود که با

مغالیت ای قبلی هم پوشانی ندارد و زودترین زمان

بدون را دارد.

ادعا: جواب الگوریتم حریصانه بهینه است.



خرف کنیم که مجموعه مغالیت های الگوریتم حریصانه

* مجموعه بهینه منته به باشند.

* جواب بهینه ای وجود دارد که تمام مغالیت های که الگوریتم حریصانه آنها را انتخاب کرده، شامل می شود.

اثبات: هر مغالیتی که در جواب حریصانه باشد یا در جواب بهینه است و یا می توانیم با مغالیتی در جواب بهینه آن را جایگزین کنیم که با هم هم پوشانی نداشته و توجه کنید زمان مغالیت در الگوریتم بهینه کوچکتر از زمان پایان مغالیت جایگزین شده حریصانه است.

→ ادامه اثبات

به نام خدا

۵:۴۰

شروع کلاس

C++

زبان های برنامه نویسی

Python

همچنین تعداد صفات آن مجموع S^* از k بیشتر است

زیرا در غیر این صورت صفات این می توانست به k اضافه
int $\pi = 0$

for ($i=1; i \leq n; i++$) {
شود.

if ($S[i] > \pi$) {
پیچیدگی $O(n \log n)$

فد را به لیست S اضافه کن.

$\pi = S[i]$
}

}
return S ;