Greedy

陳克盈 Koying

2024-08-14

貪心入門

- 什麼是貪心？

貪心入門

- 什麼是貪心？
- 人的本性？

貪心入門

- 什麼是貪心？
- 人的本性？
- 確實

貪心入門

- 什麼是貪心？
- 人的本性？
- 確實
- 有沒有更精確的描述方法？

貪心入門

- 什麼是貪心？
- 人的本性？
- 確實
- 有沒有更精確的描述方法？
- 我的答案是：不斷選擇「目前」最好的選項

來一個簡單的題目

硬幣問題 1

你有無限多個面額為 $1, 5, 10, 50$ 的硬幣，請問要怎麼用最少的硬幣數量求出 n 元？

硬幣問題

- 相信有買過東西的人應該都知道該怎麼辦
- 顯然就是先拿面額大的嘛

硬幣問題

- 相信有買過東西的人應該都知道該怎麼辦
- 顯然就是先拿面額大的嘛
- 但，這個策略難道永遠都是正確的嗎？

硬幣問題 2

你有無限多個面額為 $1, 5, 11$ 的硬幣，請問要怎麼用最少的硬幣數量求出 n 元？

- 使用剛剛的策略還會正確嗎？

硬幣問題 2

你有無限多個面額為 $1, 5, 11$ 的硬幣，請問要怎麼用最少的硬幣數量求出 n 元？

- 使用剛剛的策略還會正確嗎？
- 假設 $n = 15$，使用剛剛的策略會拿 $11 + 1 + 1 + 1 + 1$
- 但是 $5 + 5 + 5$ 才是最少的硬幣數量

硬幣問題

硬幣問題 2

你有無限多個面額為 $1, 5, 11$ 的硬幣，請問要怎麼用最少的硬幣數量求出 n 元？

- 使用剛剛的策略還會正確嗎？
- 假設 $n = 15$，使用剛剛的策略會拿 $11 + 1 + 1 + 1 + 1$
- 但是 $5 + 5 + 5$ 才是最少的硬幣數量
- 代表著這個策略並不是永遠正確的，或許只有在硬幣面額呈倍數時才成立

硬幣問題

硬幣問題 2

你有無限多個面額為 $1, 5, 11$ 的硬幣，請問要怎麼用最少的硬幣數量求出 n 元？

- 使用剛剛的策略還會正確嗎？
- 假設 $n = 15$，使用剛剛的策略會拿 $11 + 1 + 1 + 1 + 1$
- 但是 $5 + 5 + 5$ 才是最少的硬幣數量
- 代表著這個策略並不是永遠正確的，或許只有在硬幣面額呈倍數時才成立
- 那麼該如何證明呢？以下會介紹一些在離散數學中比較常見的證明方式

貪心法的常見證明方式

反證法

- proof by contradiction
- 給出某命題 p 與 $\bar{\mathrm{p}}$（非 p），其滿足排中率（$(\mathrm{p} \vee \neg\mathrm{p})$ 為真，也就是 p 與非 p 至少有一為真）

反證法

- proof by contradiction
- 給出某命題 p 與 $\bar{\mathrm{p}}$ （非 p），其滿足排中率（$(\mathrm{p} \vee \neg \mathrm{p})$ 為真，也就是 p 與非 p 至少有一為真）
- 假設 $\bar{\mathrm{p}}$ 成立，但經過推導後我們發現 $\bar{\mathrm{p}}$ 並不成立，那麼就代表 p 成立

反證法

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{p}{q}$）

- 這個反證法的例子大家高中應該都有學過

反證法

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{\mathrm{p}}{\mathrm{q}}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{\mathrm{p}}{\mathrm{q}}$

反證法

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{\mathrm{p}}{\mathrm{q}}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{\mathrm{p}}{\mathrm{q}}$
- 移項之後：$2\mathrm{q}^2 = \mathrm{p}^2$

反證法

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{p}{q}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{p}{q}$
- 移項之後：$2q^2 = p^2$
- 則 $2 \mid p^2$，又因此可推導出 $2 \mid p$，所以我們可以將 p 寫成 $2k$

反證法

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{p}{q}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{p}{q}$
- 移項之後：$2q^2 = p^2$
- 則 $2 \mid p^2$，又因此可推導出 $2 \mid p$，所以我們可以將 p 寫成 $2k$
- 再套回去剛剛的式子：$q^2 = 2k^2$，因此 $2 \mid q$

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{p}{q}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{p}{q}$
- 移項之後：$2q^2 = p^2$
- 則 $2 \mid p^2$，又因此可推導出 $2 \mid p$，所以我們可以將 p 寫成 $2k$
- 再套回去剛剛的式子：$q^2 = 2k^2$，因此 $2 \mid q$
- 經過上面的推導，發現 p, q 皆為偶數，但是 p, q 互質，所以 p, q 不能同時為偶數，矛盾

$\sqrt{2}$ 為無理數的證明

請證明 $\sqrt{2}$ 為無理數（無法使用兩互質整數 p, q 將其表示為 $\frac{p}{q}$）

- 這個反證法的例子大家高中應該都有學過
- 假設 $\sqrt{2}$ 為有理數，那麼根據有理數的性質，$\sqrt{2} = \frac{p}{q}$
- 移項之後：$2q^2 = p^2$
- 則 $2 \mid p^2$，又因此可推導出 $2 \mid p$，所以我們可以將 p 寫成 $2k$
- 再套回去剛剛的式子：$q^2 = 2k^2$，因此 $2 \mid q$
- 經過上面的推導，發現 p, q 皆為偶數，但是 p, q 互質，所以 p, q 不能同時為偶數，矛盾
- 因此 $\sqrt{2}$ 為無理數

一些反證法的經典問題

群體中，認識人數問題

試證明在一個 n 人的群體中，至少有兩個人認識的人數一樣

算幾不等式的證明

已知 $a, b \in \mathbb{R}^+$（正實數），證明 $\frac{a+b}{2} \geq \sqrt{ab}$

數學歸納法的步驟還蠻簡單的：

1. 證明 $n = 1$ 時成立
2. 當 $n = m$ 時，證明 $n = m + 1$ 時成立

有點類似我推倒第一張骨牌，接下來每一張骨牌都會因前一張骨牌倒下而倒下

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立
2. 當 $n = m$ 時，$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$ 成立

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立
2. 當 $n = m$ 時，$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$ 成立
3. 則 $n = m + 1$ 時，$\sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + (m+1)^2$

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立
2. 當 $n = m$ 時，$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$ 成立
3. 則 $n = m + 1$ 時，$\sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + (m+1)^2$
4. $\Rightarrow \sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + \frac{6(m+1)(m+1)}{6}$

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立
2. 當 $n = m$ 時，$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$ 成立
3. 則 $n = m + 1$ 時，$\sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + (m+1)^2$
4. $\Rightarrow \sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + \frac{6(m+1)(m+1)}{6}$
5. $\Rightarrow = \frac{(m+1)(m+2)(2m+3)}{6}$，滿足 $n = m + 1$ 時的式子

平方和公式

試證明 $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

1. $n = 1$ 時，成立
2. 當 $n = m$ 時，$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$ 成立
3. 則 $n = m + 1$ 時，$\sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + (m+1)^2$
4. $\Rightarrow \sum_{i=1}^{m} i^2 + (m+1)^2 = \frac{m(m+1)(2m+1)}{6} + \frac{6(m+1)(m+1)}{6}$
5. $\Rightarrow = \frac{(m+1)(m+2)(2m+3)}{6}$，滿足 $n = m + 1$ 時的式子
6. 由數學歸納法得證

數學歸納法

證明費式數列 $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

$$F(x) = \begin{cases} 0 & x = 0 \\ 1 & x \leq 1 \\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

1. $n = 0$ 時，$F_0 = F_2 - 1$，成立

證明費式數列 $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

$$F(x) = \begin{cases} 0 & x = 0 \\ 1 & x \leq 1 \\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

1. $n = 0$ 時，$F_0 = F_2 - 1$，成立
2. 假設 $n = m$ 時成立，此時 $\sum_{i=0}^{m} F_i = F_{m+2} - 1$ 成立

數學歸納法

證明費式數列 $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

$$F(x) = \begin{cases} 0 & x = 0 \\ 1 & x \leq 1 \\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

1. $n = 0$ 時，$F_0 = F_2 - 1$，成立
2. 假設 $n = m$ 時成立，此時 $\sum_{i=0}^{m} F_i = F_{m+2} - 1$ 成立
3. 在 $n = m + 1$ 時，可得 $\sum_{i=0}^{m+1} F_i = F_{m+2} - 1 + F_{m+1}$

數學歸納法

證明費式數列 $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

$$F(x) = \begin{cases} 0 & x = 0 \\ 1 & x \leq 1 \\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

1. $n = 0$ 時，$F_0 = F_2 - 1$，成立
2. 假設 $n = m$ 時成立，此時 $\sum_{i=0}^{m} F_i = F_{m+2} - 1$ 成立
3. 在 $n = m + 1$ 時，可得 $\sum_{i=0}^{m+1} F_i = F_{m+2} - 1 + F_{m+1}$
4. 可得 $\sum_{i=0}^{m+1} F_i = F_{m+3} - 1$

數學歸納法

證明費式數列 $\sum_{i=0}^{n} F_i = F_{n+2} - 1$

$$F(x) = \begin{cases} 0 & x = 0 \\ 1 & x \leq 1 \\ f(x-1) + f(x-2) & x > 1 \end{cases}$$

1. $n = 0$ 時，$F_0 = F_2 - 1$，成立
2. 假設 $n = m$ 時成立，此時 $\sum_{i=0}^{m} F_i = F_{m+2} - 1$ 成立
3. 在 $n = m + 1$ 時，可得 $\sum_{i=0}^{m+1} F_i = F_{m+2} - 1 + F_{m+1}$
4. 可得 $\sum_{i=0}^{m+1} F_i = F_{m+3} - 1$
5. 由數學歸納法得證

構造性證明

- 利用建立出一種構造方式，證明假設正確

構造性證明

證明質數有無限多個

請證明質數有無限多個

- 假設已知質數有 k 個 $p_1, p_2, \ldots, p_k$

構造性證明

證明質數有無限多個

請證明質數有無限多個

- 假設已知質數有 k 個 $p_1, p_2, \ldots, p_k$
- 那麼我們就可以構造出一個數 $x = \prod_1^k p_k + 1$

證明質數有無限多個

請證明質數有無限多個

- 假設已知質數有 k 個 $p_1, p_2, \ldots, p_k$
- 那麼我們就可以構造出一個數 $x = \prod_1^k p_k + 1$
- 如果 x 為和數，那麼一定可以找到一個質因數 P 使得 $P \mid x$

構造性證明

證明質數有無限多個

請證明質數有無限多個

- 假設已知質數有 k 個 $p_1, p_2, \ldots, p_k$
- 那麼我們就可以構造出一個數 $x = \prod_1^k p_k + 1$
- 如果 x 為和數，那麼一定可以找到一個質因數 P 使得 $P \mid x$
- 但這是不可能的，因為 $x \equiv 1 \pmod{p_i}$ 所以，x 為質數，得證

回到硬幣問題

硬幣問題

你有無限多個面額為 $c_1, c_2, \ldots, c_n$ 的硬幣，請問要怎麼用最少的硬幣數量求出 x 元？
已知 $v_1 = 1, v_i \mid v_{i+1}$（$v_i$ 能夠整除 v_{i+1}）

1. 假設對於某個 c_i 的硬幣，我們用了超過 $\frac{c_{i+1}}{c_i}$ 個，那麼我們就可以將這 $\frac{c_{i+1}}{c_i}$ 個 c_i 換成是 c_{i+1}

回到硬幣問題

硬幣問題

你有無限多個面額為 $c_1, c_2, \ldots, c_n$ 的硬幣，請問要怎麼用最少的硬幣數量求出 x 元？
已知 $v_1 = 1, v_i \mid v_{i+1}$ （v_i 能夠整除 v_{i+1}）

1. 假設對於某個 c_i 的硬幣，我們用了超過 $\frac{c_{i+1}}{c_i}$ 個，那麼我們就可以將這 $\frac{c_{i+1}}{c_i}$ 個 c_i 換成是 c_{i+1}
2. 根據以上所述，每個硬幣的數量都不會超過 $\frac{c_{i+1}}{c_i}$ 個，且在最佳解中，若只使用 $c_1 \sim c_{i-1}$ 的硬幣，所能湊出的最大金額為 $v_i - 1$

回到硬幣問題

硬幣問題

你有無限多個面額為 $c_1, c_2, \ldots, c_n$ 的硬幣，請問要怎麼用最少的硬幣數量求出 x 元？
已知 $v_1 = 1, v_i \mid v_{i+1}$ （v_i 能夠整除 v_{i+1}）

1. 假設對於某個 c_i 的硬幣，我們用了超過 $\frac{c_{i+1}}{c_i}$ 個，那麼我們就可以將這 $\frac{c_{i+1}}{c_i}$ 個 c_i 換成是 c_{i+1}
2. 根據以上所述，每個硬幣的數量都不會超過 $\frac{c_{i+1}}{c_i}$ 個，且在最佳解中，若只使用 $c_1 \sim c_{i-1}$ 的硬幣，所能湊出的最大金額為 $v_i - 1$
3. 因此當我們需要求出 x 元，且 $c_i \leq x < c_{i+1}$ 時，必選 c_i，得證在面額有倍數關係時，貪心解為最佳解

貪心經典題

一些很簡單的問題

CSES Maximum Subarray Sum

求最大子陣列和

- 還記得區間和怎麼算嗎？

一些很簡單的問題

CSES Maximum Subarray Sum

求最大子陣列和

- 還記得區間和怎麼算嗎？
- $\sum_{i=l}^{r} a_i = psum_r - psum_{l-1}$

一些很簡單的問題

CSES Maximum Subarray Sum

求最大子陣列和

- 還記得區間和怎麼算嗎？
- $\sum_{i=l}^{r} a_i = psum_r - psum_{l-1}$
- 對於 r 為結尾的最大子陣列和，顯然就是一個最小的 $psum_i$ 滿足 $i < r$

一些很簡單的問題

CSES Maximum Subarray Sum

求最大子陣列和

- 還記得區間和怎麼算嗎？
- $\sum_{i=l}^{r} a_i = psum_r - psum_{l-1}$
- 對於 r 為結尾的最大子陣列和，顯然就是一個最小的 $psum_i$ 滿足 $i < r$
- 那麼對於整個陣列的最大子陣列和，答案就是 $\max_{r=1}^{n}(psum_r - \max_{i=1}^{r-1}(i))$

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

- 觀察題目，發現可能會有兩種方式：依照開頭排序、依照結尾排序

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

- 觀察題目，發現可能會有兩種方式：依照開頭排序、依照結尾排序
- 以開頭排序的情況，很簡單的可以發現，若有一部特別早開始的電影，但是特別晚結束，那麼這部電影的時間就很有可能包含了其他部電影，導致不會是最佳解

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

- 觀察題目，發現可能會有兩種方式：依照開頭排序、依照結尾排序
- 以開頭排序的情況，很簡單的可以發現，若有一部特別早開始的電影，但是特別晚結束，那麼這部電影的時間就很有可能包含了其他部電影，導致不會是最佳解
- 那如果以結尾排序呢？假設我們目前在 x 之後有空，那麼根據策略看了一部最早結束的電影 i 之後，變為 b_i 之後有空。對於任意一個解所看的電影 j 看完後有空的時間變為 b_j，b_i 必定 $\leq b_j$

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

- 觀察題目，發現可能會有兩種方式：依照開頭排序、依照結尾排序
- 以開頭排序的情況，很簡單的可以發現，若有一部特別早開始的電影，但是特別晚結束，那麼這部電影的時間就很有可能包含了其他部電影，導致不會是最佳解
- 那如果以結尾排序呢？假設我們目前在 x 之後有空，那麼根據策略看了一部最早結束的電影 i 之後，變為 b_i 之後有空。對於任意一個解所看的電影 j 看完後有空的時間變為 b_j，b_i 必定 $\leq b_j$
- 顯然在剩下的時間內，b_i 之後所能看的電影數量必定 $\geq b_j$

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

- 觀察題目，發現可能會有兩種方式：依照開頭排序、依照結尾排序
- 以開頭排序的情況，很簡單的可以發現，若有一部特別早開始的電影，但是特別晚結束，那麼這部電影的時間就很有可能包含了其他部電影，導致不會是最佳解
- 那如果以結尾排序呢？假設我們目前在 x 之後有空，那麼根據策略看了一部最早結束的電影 i 之後，變為 b_i 之後有空。對於任意一個解所看的電影 j 看完後有空的時間變為 b_j，b_i 必定 $\leq b_j$
- 顯然在剩下的時間內，b_i 之後所能看的電影數量必定 $\geq b_j$
- 得證不存在另解優於貪心解 $\Rightarrow$ 貪心解為最佳解

例題

CSES Movie Festival

有 n 場電影，每場電影從 a_i 到 b_i，請問最多可以看幾場電影？$(n \leq 2 \cdot 10^5)$

CSES Movie Festival II（APCS 202301 P4 加強版）

同 Movie Festival，但是共有 k 人能同時看電影，求最多能看幾場電影？(重複不算)

TIOJ 1072 誰先晚餐

有 n 個人，第 i 個人的餐點需要做 C_i 分鐘、需要吃 E_i 分鐘。廚師一次只能做一道菜，但每個人可以同時吃東西，請問最少需要多少時間才能讓所有人吃完？

ZJ b231. TOI2009 第三題：書

同 TIOJ 1072

例題

CSES Room Allocation

有 n 為客人，第 i 位客人的入住時間為 a_i，退房時間為 b_i，請問最多需要幾間房間來安排客人？並構造出分配方法

赫夫曼編碼

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 假設 $n = 1, 2$，那麼答案很顯然

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 假設 $n = 1, 2$，那麼答案很顯然
- 那我們來看 $n = 3$ 的情況：合併會經過兩次，而最後一次的費用一定是 $\sum a_i$

赫夫曼編碼

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 假設 $n = 1, 2$，那麼答案很顯然
- 那我們來看 $n = 3$ 的情況：合併會經過兩次，而最後一次的費用一定是 $\sum a_i$
- 那麼如何讓第一次合併最小便是我們的目標，顯然就是拿兩個最小的合併

赫夫曼編碼

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 假設 $n = 1, 2$，那麼答案很顯然
- 那我們來看 $n = 3$ 的情況：合併會經過兩次，而最後一次的費用一定是 $\sum a_i$
- 那麼如何讓第一次合併最小便是我們的目標，顯然就是拿兩個最小的合併
- 我們可以猜測，最佳的貪心策略為：每次拿最小的兩個合併

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 假設 $n = 1, 2$，那麼答案很顯然
- 那我們來看 $n = 3$ 的情況：合併會經過兩次，而最後一次的費用一定是 $\sum a_i$
- 那麼如何讓第一次合併最小便是我們的目標，顯然就是拿兩個最小的合併
- 我們可以猜測，最佳的貪心策略為：每次拿最小的兩個合併
- 如何證明？

赫夫曼編碼

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 看起來這題很欠數學歸納法

赫夫曼編碼

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 看起來這題很欠數學歸納法
- 顯然 $n \leq 2$ 時一定是對的

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 看起來這題很欠數學歸納法
- 顯然 $n \leq 2$ 時一定是對的
- 假設 $n = k$ 時策略正確，又 $n = k + 1$ 可由合併任意兩個節點得到 $n = k$ 的情況

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 看起來這題很欠數學歸納法
- 顯然 $n \leq 2$ 時一定是對的
- 假設 $n = k$ 時策略正確，又 $n = k + 1$ 可由合併任意兩個節點得到 $n = k$ 的情況
- 那麼最佳解即是取兩個最小的元素合併，符合貪心策略

CSES Stick Divisions

你有 n 根棍子 $a_1, a_2, \ldots, a_n$，每次操作可以任選兩根棍子 a_i, a_j，並付出 $a_i + a_j$ 的費用將其合為一根，求最少需要多少費用才能將所有棍子合為一根？

- 看起來這題很欠數學歸納法
- 顯然 $n \leq 2$ 時一定是對的
- 假設 $n = k$ 時策略正確，又 $n = k + 1$ 可由合併任意兩個節點得到 $n = k$ 的情況
- 那麼最佳解即是取兩個最小的元素合併，符合貪心策略
- 由數學歸納法得證，貪心法正確

赫夫曼編碼

- 這個貪心策略便是鼎鼎大名的「赫夫曼編碼 Huffman Coding」
- 合併過程產生的赫夫曼樹，又被稱為最優二元樹
- 有興趣的可以上網查一下相關資料

交換貪心

交換貪心

- 其實前面的線段貪心也是交換貪心的一種
- 利用先構造出一個解，再嘗試證明出透過某種交換方式可以得到更好的解
- 最後再利用得出的交換方式來排序

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較
- 那麼所得的獎勵為：$(d_i - f_i) + (d_j - f_j)$，也就是 $(d_i - f_i) + (d_j - f_i - a_j)$

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較
- 那麼所得的獎勵為：$(d_i - f_i) + (d_j - f_j)$，也就是 $(d_i - f_i) + (d_j - f_i - a_j)$
- 也就是 $(d_i + d_j) - a_j - 2f_i$

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較
- 那麼所得的獎勵為：$(d_i - f_i) + (d_j - f_j)$，也就是 $(d_i - f_i) + (d_j - f_i - a_j)$
- 也就是 $(d_i + d_j) - a_j - 2f_i$
- 假設第 i 個任務開始的時間為 s_i，那麼式子可以改寫為：$(d_i + d_j) - 2s_i - 2a_i - a_j$

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較
- 那麼所得的獎勵為：$(d_i - f_i) + (d_j - f_j)$，也就是 $(d_i - f_i) + (d_j - f_i - a_j)$
- 也就是 $(d_i + d_j) - a_j - 2f_i$
- 假設第 i 個任務開始的時間為 s_i，那麼式子可以改寫為：$(d_i + d_j) - 2s_i - 2a_i - a_j$
- 我們可以發現到，不管 i, j 的先後順序為何，式子中 $(d_i + d_j) - 2s_i - a_i - a_j$ 都是不變的

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 首先我們先假設有一個任意解 S_1，並拿出任一組相鄰的元素 i, j $(j = i + 1)$ 來比較
- 那麼所得的獎勵為：$(d_i - f_i) + (d_j - f_j)$，也就是 $(d_i - f_i) + (d_j - f_i - a_j)$
- 也就是 $(d_i + d_j) - a_j - 2f_i$
- 假設第 i 個任務開始的時間為 s_i，那麼式子可以改寫為：$(d_i + d_j) - 2s_i - 2a_i - a_j$
- 我們可以發現到，不管 i, j 的先後順序為何，式子中 $(d_i + d_j) - 2s_i - a_i - a_j$ 都是不變的
- 唯一會變的就只有 a_i，因此我們可以得出一個策略：排前面的 a 越小，答案就越大

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 那麼該如何證明呢？

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 那麼該如何證明呢？
- 假設利用貪心法得到的解為 S，那麼這個 S 會有一個性質：$a_i \leq a_j,\ i < j$

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 那麼該如何證明呢？
- 假設利用貪心法得到的解為 S，那麼這個 S 會有一個性質：$a_i \leq a_j,\ i < j$
- 如果這個解並不是最佳解，那麼代表會有一個真正的最佳解 S' 滿足 $a_i > a_j,\ i < j$

交換貪心

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 那麼該如何證明呢？
- 假設利用貪心法得到的解為 S，那麼這個 S 會有一個性質：$a_i \leq a_j,\ i < j$
- 如果這個解並不是最佳解，那麼代表會有一個真正的最佳解 S' 滿足 $a_i > a_j,\ i < j$
- 但根據剛剛的式子推出可以發現，當 $a_i > a_j$ 時，獎勵會變得更少，因此 S' 必定 $\leq S$，S' 不滿足最佳解

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

- 那麼該如何證明呢？
- 假設利用貪心法得到的解為 S，那麼這個 S 會有一個性質：$a_i \leq a_j,\ i < j$
- 如果這個解並不是最佳解，那麼代表會有一個真正的最佳解 S' 滿足 $a_i > a_j,\ i < j$
- 但根據剛剛的式子推出可以發現，當 $a_i > a_j$ 時，獎勵會變得更少，因此 S' 必定 $\leq S$，S' 不滿足最佳解
- 根據反證法證明 S 為最佳解

例題

CSES Tasks and Deadlines

你有 n 個任務，每個任務都有兩個參數 a, d，代表完成這個任務的執行時間以及截止時間。對於每個任務，你所獲得的獎勵為 $d - f$（f 代表這個任務最終完成的時間），你每次只能做一個任務，請問最大的獎勵總和是多少？

字串的交換貪心

你有 n 個字串，請將這些字串以某種順序接在一起，使得最終的字典序最小

TPR 20G. 隊伍編排 (Permutation)

請見原題

數學貪心

CF 1175D. Array Splitting

有一個數列 $a_1, a_2, \ldots, a_n$, $(n \leq 3 \cdot 10^5)$，你可以將它分成 k 個子陣列，每個元素貢獻的分數為 $a_i \cdot f(i)$（$f(i)$ 代表第 i 個元素位處第幾個子陣列）
求最大的分數總和

- 假設第 i 個子陣列為 $[l_i, r_i]$，那麼答案可以寫成 max $\sum_{j=1}^{k} suf(l_i)$（$suf(i)$：後綴和，代表 $i \sim n$ 的和）

CF 1175D. Array Splitting

有一個數列 $a_1, a_2, \ldots, a_n$,$(n \leq 3 \cdot 10^5)$,你可以將它分成 k 個子陣列,每個元素貢獻的分數為 $a_i \cdot f(i)$($f(i)$ 代表第 i 個元素位處第幾個子陣列)
求最大的分數總和

- 假設第 i 個子陣列為 $[l_i, r_i]$,那麼答案可以寫成 max $\sum_{j=1}^{k} \mathrm{suf}(l_i)$($\mathrm{suf}(i)$:後綴和,代表 $i \sim n$ 的和)
- 不難發現,最終答案就是從 n 種後綴和中挑出 k 個

CF 1175D. Array Splitting

有一個數列 $a_1, a_2, \ldots, a_n$, $(n \leq 3 \cdot 10^5)$，你可以將它分成 k 個子陣列，每個元素貢獻的分數為 $a_i \cdot f(i)$（$f(i)$ 代表第 i 個元素位處第幾個子陣列）
求最大的分數總和

- 假設第 i 個子陣列為 $[l_i, r_i]$，那麼答案可以寫成 max $\sum_{j=1}^{k} suf(l_i)$（$suf(i)$：後綴和，代表 $i \sim n$ 的和）
- 不難發現，最終答案就是從 n 種後綴和中挑出 k 個
- 顯然貪心策略便是前 k 大的後綴和相加

反悔貪心

反悔貪心

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \leq 2 \cdot 10^5, -10^9 \leq a_i \leq 10^9)$
你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \geq 0$ 時一定可以喝

反悔貪心

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \leq 2 \cdot 10^5, -10^9 \leq a_i \leq 10^9)$ 你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \geq 0$ 時一定可以喝
- 但當 $a_i < 0$ 呢？如果喝了還不會死，那當然喝了最好

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \le 2 \cdot 10^5, -10^9 \le a_i \le 10^9)$
你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \ge 0$ 時一定可以喝
- 但當 $a_i < 0$ 呢？如果喝了還不會死，那當然喝了最好
- 但如果會死呢？

反悔貪心

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \le 2 \cdot 10^5, -10^9 \le a_i \le 10^9)$ 你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \ge 0$ 時一定可以喝
- 但當 $a_i < 0$ 呢？如果喝了還不會死，那當然喝了最好
- 但如果會死呢？
- 程式跟人生不一樣，是可以反悔的。我們可以將之前喝的其他瓶藥水改為不喝，那或許就能喝下這瓶藥水了

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \leq 2 \cdot 10^5, -10^9 \leq a_i \leq 10^9)$
你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \geq 0$ 時一定可以喝
- 但當 $a_i < 0$ 呢？如果喝了還不會死，那當然喝了最好
- 但如果會死呢？
- 程式跟人生不一樣，是可以反悔的。我們可以將之前喝的其他瓶藥水改為不喝，那或許就能喝下這瓶藥水了
- 而選擇哪瓶不喝的策略很簡單，顯然就是扣最多血的不要喝

反悔貪心

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \leq 2 \cdot 10^5, -10^9 \leq a_i \leq 10^9)$ 你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

- 首先，很顯然的，$a_i \geq 0$ 時一定可以喝
- 但當 $a_i < 0$ 呢？如果喝了還不會死，那當然喝了最好
- 但如果會死呢？
- 程式跟人生不一樣，是可以反悔的。我們可以將之前喝的其他瓶藥水改為不喝，那或許就能喝下這瓶藥水了
- 而選擇哪瓶不喝的策略很簡單，顯然就是扣最多血的不要喝
- 使用 priority queue or set 來維護

例題

CF 1526C2. Potions (Hard Version)

路上總共有 n 瓶藥水，每瓶會對你的血量造成 a_i 點的改變。$(n \le 2 \cdot 10^5, -10^9 \le a_i \le 10^9)$
你一開始有 0 點血量，你可以選擇每瓶藥水是喝還是不喝，問在血量非負的情況下，最多可以喝幾瓶藥水？

CF 1779C. Least Prefix Sum

有一個數列 $a_1, a_2, \ldots, a_n$，你可以選擇將 a 中的任意數字 a_i 改為 $-a_i$ $(a_i := -a_i)$
請問最少要改幾個數字，才能使得對於所有 i，滿足 $\sum_{j=1}^{i} a_j \le \sum_{j=1}^{m} a_j$？$(m \le n \le 2 \cdot 10^5)$

綜合例題

CSES Reading Books

有 n 本書，每本書要花 t_i 的時間讀完。問兩個人最少要花多少時間，才能在不同時看同一本書的情況將所有書看完

- 這 n 本書中一定有其中一本書 i 是需要花最久時間的，此時會有兩種情況

綜合例題

CSES Reading Books

有 n 本書，每本書要花 t_i 的時間讀完。問兩個人最少要花多少時間，才能在不同時看同一本書的情況將所有書看完

- 這 n 本書中一定有其中一本書 i 是需要花最久時間的，此時會有兩種情況
  1. $t_i \geq \sum t - t_i$
  2. $t_i < \sum t - t_i$

綜合例題

CSES Reading Books

有 n 本書，每本書要花 t_i 的時間讀完。問兩個人最少要花多少時間，才能在不同時看同一本書的情況將所有書看完

- 這 n 本書中一定有其中一本書 i 是需要花最久時間的，此時會有兩種情況
  1. $t_i \geq \sum t - t_i$
  2. $t_i < \sum t - t_i$
- 第一種情況代表任意一人在讀第 i 本書的期間，另一人可以將其他書全部看完，因此答案 $= 2t_i$

綜合例題

CSES Reading Books

有 n 本書，每本書要花 t_i 的時間讀完。問兩個人最少要花多少時間，才能在不同時看同一本書的情況將所有書看完

- 這 n 本書中一定有其中一本書 i 是需要花最久時間的，此時會有兩種情況
  1. $t_i \geq \sum t - t_i$
  2. $t_i < \sum t - t_i$
- 第一種情況代表任意一人在讀第 i 本書的期間，另一人可以將其他書全部看完，因此答案 $= 2t_i$
- 至於第二種情況，我們如果將書以 t_i 大至小排序，第一人先看 t_1，第二人先看 t_2，必定不會需要等另外一人讀完某本書，因此時間為 $\sum t$，至於為什麼就留給大家自己思考了

綜合例題

CSES Reading Books

有 n 本書，每本書要花 t_i 的時間讀完。問兩個人最少要花多少時間，才能在不同時看同一本書的情況將所有書看完

- 這 n 本書中一定有其中一本書 i 是需要花最久時間的，此時會有兩種情況
  1. $t_i \geq \sum t - t_i$
  2. $t_i < \sum t - t_i$
- 第一種情況代表任意一人在讀第 i 本書的期間，另一人可以將其他書全部看完，因此答案 $= 2t_i$
- 至於第二種情況，我們如果將書以 t_i 大至小排序，第一人先看 t_1，第二人先看 t_2，必定不會需要等另外一人讀完某本書，因此時間為 $\sum t$，至於為什麼就留給大家自己思考了
- 因此最終的答案便是 $\max(2 \times \max_{i=1}^{n} t_i, \sum t)$

綜合例題

- CF 1552C. Maximize the Intersections
- CF 1506D. Epic Transformation
- CF 1203F1. Complete the Projects (easy version)

結語

- 貪心看似簡單，但在變化多端的題目上總是埋藏著許多梗

結語

- 貪心看似簡單，但在變化多端的題目上總是埋藏著許多梗
- 貪心題目的關鍵點往往就埋藏在某個性質中

結語

- 貪心看似簡單，但在變化多端的題目上總是埋藏著許多梗
- 貪心題目的關鍵點往往就埋藏在某個性質中
- 看到一道題目後，不妨多多猜測，或許這個擺前面會比較好？或許交換這個會更強？

結語

- 貪心看似簡單，但在變化多端的題目上總是埋藏著許多梗
- 貪心題目的關鍵點往往就埋藏在某個性質中
- 看到一道題目後，不妨多多猜測，或許這個擺前面會比較好？或許交換這個會更強？
- 「大膽假設、小心求證」，除了課堂中的題目外，也可以上 CF、CSES 找到相關 tag 的題目多多練習

結語

- 貪心看似簡單，但在變化多端的題目上總是埋藏著許多梗
- 貪心題目的關鍵點往往就埋藏在某個性質中
- 看到一道題目後，不妨多多猜測，或許這個擺前面會比較好？或許交換這個會更強？
- 「大膽假設、小心求證」，除了課堂中的題目外，也可以上 CF、CSES 找到相關 tag 的題目多多練習
- 相信大家都能夠學出心得！