

۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹
 $[1, 14, 12, 11, 2, 3, 9, 8, 13, 1]$

①

$countArray = [0]$

۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰
 $countArray = [0, 2, 1, 3, 0, 0, 0, 0, 1, 0, 0]$

۱۱ ۱۲ ۱۳ ۱۴
 $[1, 1, 1, 1]$

در مرحله به $prefix\ sum$ را به $countArray$ حساب می‌کنیم

۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴
 $countArray = [0, 2, 3, 4, 4, 4, 4, 4, 5, 4, 4, 7, 8, 9, 10]$

حال از انتهای $inputArray$ شروع می‌کنیم و برای هر اندیس i داریم:

مراحل زیر را انجام می‌دهیم:

$outputArray[i] = [countArray[inputArray[i]] - 1] = inputArray[i]$

$countArray[inputArray[i]] --$

idea (5)

output Array [count Array[input Array[9]]-1] = 2

input Array [a]

→ count Array [inputArray [a]] =

~~Let Array $E[1..N]$ = Input Array E of~~

count Array [1] = 4

$\text{outputArray}[x-1] = \text{inputArray}[a]$

output Array [1] = 1

count Array [input Array[a]] -- \rightarrow

$\text{count Array}[1] \rightarrow \text{count Array}[1] = 1$

برای $i = 1$ داریم :

$$\text{output Array}[\text{count Array}[\text{input Array}[1]] - 1] =$$

$$\text{input Array}[1]$$

$$\text{output Array}[\text{count Array}[13] - 1] \rightarrow$$

$$\text{output Array}[9 - 1] = \text{input Array}[1] = 13$$

$$\text{count Array}[\text{input Array}[1]] \dots \rightarrow$$

$$\text{count Array}[13] \dots \rightarrow \text{count Array}[13] = 1$$

به همین صورت برای $i = 7$ تا $i = 0$ هم همین مراحل را انجام

می دهیم و در هر مرحله output Array ، count Array را به روز می کنیم

و در نهایت آرایه خروجی را به دست می آوریم:

$$\text{output Array} = [1, 1, 2, 3, 1, 2, 11, 12, 13, 14]$$

② یک راه حل این است که اعداد ورودی به درجانه $[P, 9]$

با به بازه (۱ تا ۱۰۰۰) تبدیل کنیم که به اصطلاح نرغال سازی می‌گویند

از فرمول زیر نرغال برای نرغال سازی استفاده کرد:

$$\frac{x - p}{2 - p}$$

در نرغال آخری الوریتم هم از الوریتم های مرتب سازی هر bucket

از الوریتم efficient استفاده کنیم تا بهتری «لینک» پیدا

Merge sort

۳) به ی‌توانیم، دانستیم عددی اعداد را به میان n تبدیل کنیم.

یعنی به مرتبه abc به شایسته که اگر بخوانیم به میان a تبدیل

کنیم می‌تواند $ax^2 + bx + c$ حال از الگوریتم

ratio sort استفاده کنیم بصورتی که می‌دانیم حد اکثر عدد ما ۳ رقم دارد

و به کار داریم به برای مکانی هر رقم از counting sort

استفاده کنیم به مرتبه $O(n)$ (نکته) می‌برد به در زمان

خفای مرتبه می‌تواند.

$$(n^3 - 1)_1 \leq (n-1)(n-1)(n-1)_n$$

④ هر عملیات برای ستاره‌ها ستی را مقدار می‌دهد flip

می‌تواند یا به اصطلاح از آن به یا با تحلیل تغییر می‌دهند به عنوان مثال

برای ۵ تا ۴ داریم؟

عدد ستاره‌ها

مقدار مابقی

عملیات

۰	۰۰۰۰
۱	۱۰۰۰
۲	۰۰۱۰
۳	۰۰۱۱
۴	۱۰۰۰

۰	↓ +۱
۱	↓ +۲
۲	↓ +۱
۳	↓ +۳

تحلیل غیر سرشکن: در بدترین حالت فرض می‌کنیم در هر increment

همیشه ما باید flip کنیم به از مرتبه $O(\log n)$ است

تکرار از ۰ شروع کنیم تا به عدد n برسیم، باید n بار این عملیات

را تکرار کنیم به از مرتبه $O(n)$ است به درجه از مرتبه

$O(n \log n)$ است.

⑤ برای شمار ریاضی از دو زیر گروه ۲۸ عضوی جدا شده

X مردان n عضو انتخاب شد. تعداد آن ها طبق ریاضیات تعیین

برای به دست آوردن

$$\binom{2n}{n}^2 = \left(\frac{(2n)!}{n! \cdot n!} \right)^2$$

④

stable

زوال اجرای متد

بترین زوال اجرا

الگوریتم

✓

$O(n^2)$

$O(n^2)$

insertion

✓

$O(n \log n)$

$O(n \log n)$

Merge

✗

$O(n \log n)$

$O(n \log n)$

Heap

✗

$O(n \log n)$

$O(n^2)$

Quick

✓

$O(n+m)$

$O(n+m)$

Counting

✓

$O(d(n+b))$

$O(d(n+b))$

Radix

✓

$O(n)$

$O(n^2)$

Bucket