14-C3

排序

希尔排序：PS序列

邓俊辉

deng@tsinghua.edu.cn
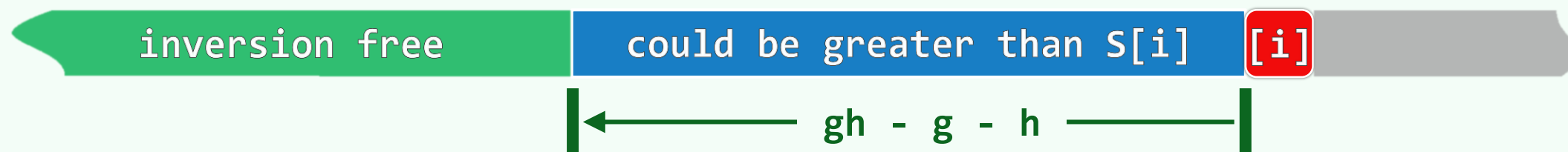
d-Sorting an $\mathcal{O}$(d)-Ordered Sequence in $\mathcal{O}$(dn) Time

❖ **If g and h are relatively prime and are both in** $\mathcal{O}(d)$

we can d-sort the sequence in $\mathcal{O}(dn)$ **time ...**

- **re-arrange the sequence as a 2D matrix with d columns**

- **each element is swapped with** $\mathcal{O}((g-1) \cdot (h-1)/d) = \mathcal{O}(d)$ **elements**

❖ **Since this holds for all elements,** $\mathcal{O}(dn)$ **steps are enough**

| inversion free | could be greater than S[i] | [i] |

gh - g - h

PS Sequence

❖ **Papernov & Stasevic, 1965** **//also called Hibbard's sequence**

$$\mathcal{H}_{PS} = \mathcal{H}_{Shell} - 1 = \{\, 2^k - 1 \mid k \in \mathcal{N} \,\} = \{\, 1,\ 3,\ 7,\ 15,\ 31,\ 63,\ 127,\ 255,\ \ldots \,\}$$

❖ **Different items MAY NOT be relatively prime, e.g.,** $h_{2k} = h_k \cdot (h_k + 2)$

But ADJACENT items MUST be, since $h_{k+1} - 2 \cdot h_k \equiv 1$

❖ **Shellsort with** $\mathcal{H}_{ps}$ **needs**

- $\mathcal{O}(\log n)$ **outer iterations and**

- $\mathcal{O}(n^{3/2})$ **time to sort a sequence of length n** **//Why ...**

t < k

❖ **Let** h_t **be the** h **closest to** $\sqrt{n}$ **and hence** $h_t \approx \sqrt{n} = \Theta(n^{1/2})$

1) **Consider those iterations for** $\{\, h_k \mid t < k \,\} = \{\, \overleftarrow{h_{t+1},\ h_{t+2},\ \ldots,\ h_m}\, \}$

∵ **there would be** $\mathcal{O}(n/h_k)$ **elements in each of the** h_k **columns**

∴ **we can insertionsort each column in** $\mathcal{O}((n/h_k)^2)$ **time**

∴ **each h_k-sorting costs** $\mathcal{O}(n^2/h_k)$ **time**

∴ **all these iterations cost time of**

$$\mathcal{O}(2 \times n^2/h_t) = \mathcal{O}(n^{3/2})$$

$t \leq k$
$h_t \leq h_k$

$t < k$
$h_t < h_k$
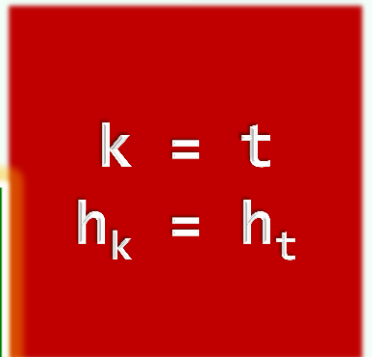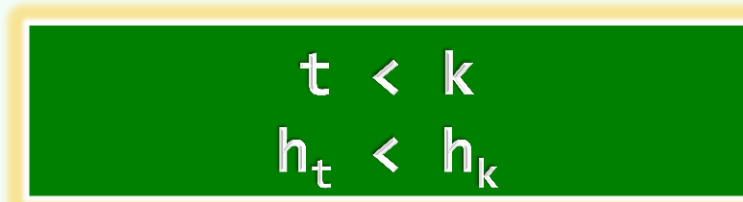
$k = t$
$h_k = h_t$

2) Consider those iterations for $\{\ h_k\ |\ k \le t\ \} = \{\ \overleftarrow{h_1,\ h_2,\ \ldots,\ h_t}\ \}$

∵ h_{k+1} and h_{k+2} are relatively prime and are both in $\mathcal{O}(h_k)$

∴ each h_k-sorting costs $\mathcal{O}(n \times h_k)$ time

∴ all these iterations cost $\mathcal{O}(n \times 2 \cdot h_t) = \mathcal{O}(n^{3/2})$ time

❖ This upper bound is TIGHT

❖ What about the average cases?

- $\mathcal{O}(n^{5/4})$ based on simulations

- but not proved yet

$t \le k$
$h_t \le h_k$

$t < k$
$h_t < h_k$

$k = t$
$h_k = h_t$