MR 8.22 Consider any $x, y \in [u], x \neq y$. Observe that $|H| = p - 1$. As in the lecture notes, we will bound the number of a's such that $h_a(x) = h_a(y)$. Let $u = ax \bmod p, v = ay \bmod p$. Since $x \neq y$ and $a \neq 0$, we have $u \neq v$. We rewrite the collision condition:

$$\begin{aligned} h(x) &= h(y), \\ u \bmod n &= v \bmod n, \\ (u - v) \bmod n &= 0. \end{aligned}$$

Since $u, v \in [p]$ and $u \neq v$, $u - v$ can take $2(\lceil \frac{p}{n} \rceil - 1) \leq \frac{2(p-1)}{n}$ different values: $\ldots, -2n, -n, n, 2n, \ldots$. Meanwhile, since $(ax \bmod p) - (ay \bmod p) = u - v$, i.e.,

$$a(x - y) = u - v \qquad (\bmod p).$$

So for every value of $u - v$, there is one value for a such that the above equation is true. Therefore, there are at most $\frac{2(p-1)}{n}$ different values of a such that $h_a(x) = h_a(y)$. So $\Pr_{h \in H}[h(x) = h(y)] \leq \frac{2(p-1)}{n} / (p - 1) \leq 2/n$.

RIC The algorithm and analysis are very similar to those for the convex hull problem. After inserting an element x after y, we check all elements pointed by the list of pointers stored at y. For each such z, we compare z with x. If $z < x$, it stays in the pointer list at y; if $z > x$, we move it to the pointer list of x, meanwhile we update the pointer at z to point to x.

The cost of this operation is proportional to the size of the pointer list at y. Now consider the i-th insertion, i.e., when the sorted list grows from $i - 1$ elements to i elements. Using backward analysis, we view this operation as deleting one element, chosen randomly, from the i elements. For any element z yet to be inserted, it is affected by this deletion if it points to the deleted element or its predecessor (in the latter case, z needs to be checked but not moved). This happens with probability $2/i$. There are $n - i$ remaining elements, so the expected cost for the i-th insertion is $O((n - i)/i)$. Summing up all these costs by linearity of expectation, we obtain a total expected cost of $O(n \log n)$.

KT 13.14 We randomly and independently assign each process to the two machines with probability 1/2 each. Then we check if all jobs are nearly balanced, and repeat the process if not. It is obvious that each trial of this algorithm takes polynomial time. Below we show that the algorithm only repeats $O(1)$ times.

Consider any job. Let X be the number of processes of this job assigned to machine M_1. We know that $\mu = E[X] = n$. The job is not balanced iff $X < \frac{2}{3}n$ or $X > \frac{4}{3}n$. By Chernoff inequality, this happens with probability

$$\Pr\left[|X - \mu| > \frac{1}{3}\mu\right] < 2\exp\left(-\left(\frac{1}{3}\right)^2 / 3 \cdot n\right) < \frac{1}{2n},$$

where the last inequality holds for n larger than some constant. Then by the union bound, the probability that some job is not balanced is at most 1/2. So one trial fails with probability at most 1/2, thus we expect to have at most 2 trials before the algorithm succeeds.

KT 13.15 Let x_1 be the $(\frac{1}{2} - \varepsilon)n$-th smallest number in S and x_2 the $(\frac{1}{2} + \varepsilon)n$-th smallest number in S. Let X be the random variable denoting the number of numbers in the sample that are smaller than x_1. The returned median is smaller than x_1 iff $X > k/2$. Let $X_i =$ if the i-th sampled number is smaller than x_1 and 0 otherwise. We have $E[X_i] = \frac{1}{2} - \varepsilon$, $X = \sum_{i=1} kX_i$, and $E[X] = \mu = (\frac{1}{2} - \varepsilon)k$. By the Chernoff inequality,

$$\Pr[X > k/2] = \Pr[X > \mu + \varepsilon k] \leq \Pr[X > (1 + 2\varepsilon)\mu] < \exp\left(-\frac{\mu(2\varepsilon)^2}{3}\right) \leq e^{-\varepsilon^2 k/3},$$

where the last inequality holds as long as $\varepsilon < 1/4$.

Similarly, redefine X to be the random variable denoting the number of numbers in the sample that are greater than x_2. The returned median is greater than x_2 iff $X > k/2$. The same analysis shows that this happens also with probability at most $e^{-\varepsilon^2 k/3}$. Thus, by the union bound, the probability that the returned median is not an ε-approximate median is at most $2e^{-\varepsilon^2 k/3}$. If we need this probability to be δ, we need

$$2e^{-\varepsilon^2 k/3} = \delta \quad \Leftrightarrow \quad -\varepsilon^2 k/3 = \ln(\delta/2) \quad \Leftrightarrow \quad \varepsilon^2 k/3 = \ln\frac{2}{\delta} \quad \Leftrightarrow \quad k = O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right).$$

If we use a pairwise independent hash function to sample, then we can only use Chebyshev inequality on X. Since $Var[X_i] = (\frac{1}{2} - \varepsilon)(\frac{1}{2} + \varepsilon) \leq 1/4$, $Var[X] \leq k/4$, $\sigma_X = \sqrt{k}/2$, we have

$$\Pr[X > k/2] = \Pr[X - \mu > \varepsilon k] = \Pr[X - \mu > 2\varepsilon\sqrt{k}\sigma_X] = \frac{1}{4k\varepsilon^2}.$$

Again by the union bound, the the probability that the returned median is not an ε-approximate median is at most $\frac{1}{2k\varepsilon^2}$. If we need this probability to be δ, we need

$$\frac{1}{2k\varepsilon^2} = \delta \quad \Leftrightarrow \quad k = \frac{1}{2\varepsilon^2\delta}.$$

So, the relationship between k and ε is the same as in the full independent case, but that between k is δ is not. This result again tells us that full independence boosts confidence $(1 - \delta)$, but pairwise independence is sufficient for error (ε).