首先计算 $T(n,k)$.

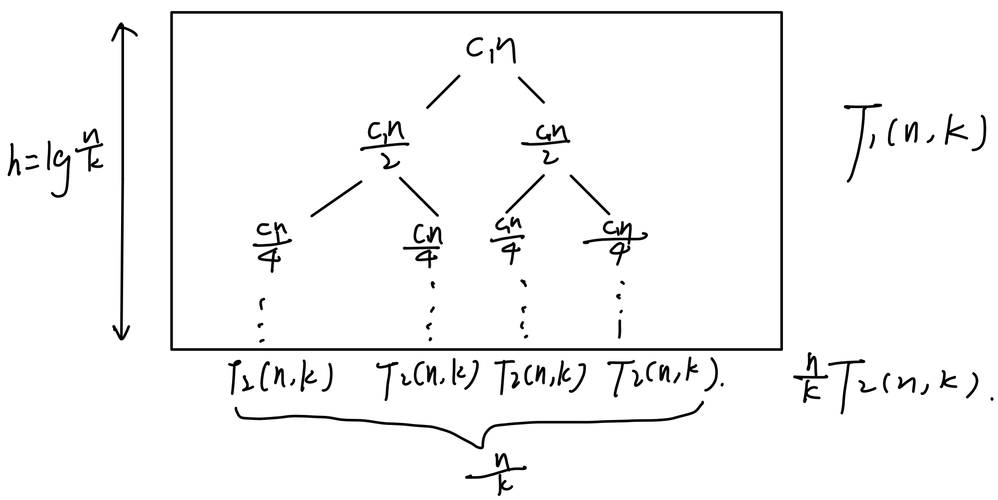
由于先使用 merge sort 再对小子数组使用 insertion sort

故 $T(n,k) = T_1(n,k) + T_2(n,k)$，其中 $T_1(n,k)$ 为 merge sort 部分

$T_2(n,k)$ 为 insertion sort 部分

假设 n, k 均为2的次方.

$$T_1(n,k) = \begin{cases} \theta(1)，当 n=k时 (调用 insertion sort) \\ 2T(\frac{n}{2}) + \theta(n)，当 n>k时， \end{cases}$$



$\therefore T_1(n,k) = c_1 n\lg\frac{n}{k} = O(n\lg\frac{n}{k})$

对于 $T_2(n,k)$，即为对一个有 k 个元素的数组进行 insertion sort.

$\Rightarrow$ 根据公式有 $\frac{n}{k}T_2(n,k) = \frac{n}{k}(pk^2+qk+r) = n(pk+q+\frac{r}{k}) = O(nk)$

故 $T(n,k) = c_1 n\lg\frac{n}{k} + n(pk+q+\frac{r}{k})$

$\Rightarrow O(T(n,k)) = O(n\lg\frac{n}{k} + nk) \Rightarrow f(n,k) = n\lg\frac{n}{k} + nk$

其次分析k的选取

理论上：

即对 $T(n,k)$ 关于 k 求导

则有 $T'(n,k) = -\frac{c_1 n}{k\ln 2} + np - \frac{nr}{k^2}$

当 n, c_1, p, r 为常数时，令 $T'(n,k)=0$，此时 k 则为选取值

实验中：观察到 $T(n,k)$ 关于 k 呈先减后增，故用下坡法找 k

① 对于固定 n 与数据，选定 k 的起始值，分别为一个较大值，一个较小值作为实验范围

如 $k_1=1$，$k_2=1000$，并取一 k'，k' 为 $(1,1000)$ 范围内一随机数

② 分别记录改进后的程序在 $k'-1$，k'，$k'+1$ 情况下的运行时间进行比较（多次取平均）

③ 若 $t_{k'-1} < t_{k'} < t_{k'+1}$，则在 $(k_1, k'-1)$ 中重复随机取数并比较过程

若 $t_{k'-1} > t_{k'} > t_{k'+1}$，则在 $(k'+1, k_2)$ 中重复重复随机取数并比较过程

若 $t_{k'-1} > t_{k'}$ 且 $t_{k'+1} > t_{k'}$，则 k' 即为合适的 k 值

同时，在实验中发现，由于机器运行程序时间的波动性，有时可能出现"假坡底"情况，因此有时误差较大，故同时给出了用枚举法找 k 值的代码（见 improved.cpp）

枚举法：对于每个 $k \in [k_1, k_2]$，计录其运行时间（多次取平均），最后比较得到最佳 k。