Análisis de Complejidad de Hash Sort

Análisis detallado de la función de tiempo $T_p(n)$ para Hash Sort

Mejor caso

$$
\begin{aligned}
T_p(n) &= t_1 + t_2 + \sum_{i=1}^{n} (t_3 + t_4 + t_5 + t_6 + t_7 + t_8) + t_9 \\
T_p(n) &= t_{123} + \sum_{i=1}^{n} t_{345678} + t_9 \\
T_p(n) &= t_{123} + n \cdot t_{345678} + t_9 \\
T_p(n) &= c_1 n + c_2
\end{aligned}
$$

Donde:

$$
\begin{aligned}
c_1 &= t_{345678} \\
c_2 &= t_{123} + t_9
\end{aligned}
$$

Peor caso

$$
\begin{aligned}
T_p(n) &= t_1 + t_2 + \sum_{i=1}^{n-1}\left(t_3 + t_4 + \sum_{j=0}^{i}(t_5 + t_6 + t_7) + t_8\right) + t_9 \\
T_p(n) &= t_{123} + \sum_{i=1}^{n-1}\left(t_{3458} + \sum_{j=0}^{i} t_{567}\right) + t_9 \\
T_p(n) &= t_{123} + \sum_{i=1}^{n-1} t_{3458} + \sum_{i=1}^{n-1}\sum_{j=0}^{i} t_{567} + t_9 \\
T_p(n) &= t_{123} + (n-1)t_{3458} + \sum_{i=1}^{n-1}(i+1)t_{567} + t_9 \\
T_p(n) &= t_{123} + (n-1)t_{3458} + t_{567}\left(\sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} 1\right) + t_9 \\
T_p(n) &= t_{123} + (n-1)t_{3458} + t_{567}\left(\frac{(n-1)n}{2} + (n-1)\right) + t_9 \\
T_p(n) &= t_{123} + (n-1)t_{3458} + \left(\frac{n(n-1) + 2(n-1)}{2}\right) t_{567} + t_9 \\
T_p(n) &= t_{123} + (n-1)t_{3458} + \left(\frac{n^2 + n - 2}{2}\right) t_{567} + t_9
\end{aligned}
$$

Forma cuadrática aproximada

$$T_p(n) = c_1 n^2 + c_2 n + c_3$$

Donde:

$$
\begin{aligned}
c_1 &= \frac{t_{567}}{2} \\
c_2 &= t_{3458} + \frac{t_{567}}{2} \\
c_3 &= t_{123} - t_{3458} - t_{567} + t_9
\end{aligned}
$$

Caso promedio

$$\begin{aligned}
T_{1/2}(n) &= \frac{1}{n}\sum_{i=1}^{n} T_p(i) \\
&= \frac{1}{n}\sum_{i=1}^{n}(c_1 i^2 + c_2 i + c_3) \\
&= \frac{1}{n}\left(c_1\sum_{i=1}^{n} i^2 + c_2\sum_{i=1}^{n} i + c_3\sum_{i=1}^{n} 1\right) \\
&= \frac{1}{n}\left(c_1 \cdot \frac{n(n+1)(2n+1)}{6} + c_2 \cdot \frac{n(n+1)}{2} + c_3 \cdot n\right)
\end{aligned}$$

Forma cuadrática promedio

$$T_{1/2}(n) = k_1 n^2 + k_2 n + k_3$$

Donde:

$$\begin{aligned}
k_1 &= \frac{c_1}{3} \\
k_2 &= \frac{c_1}{2} + \frac{c_2}{2} \\
k_3 &= \frac{c_1}{6} + \frac{c_2}{2} + c_3
\end{aligned}$$

Conclusión

La complejidad del algoritmo *Hash Sort* varía dependiendo de las colisiones y la función hash utilizada. En el **mejor caso**, el tiempo de ejecución es lineal:

$$\boxed{T(n) = \mathcal{O}(n)}$$

En el **caso promedio** y el **peor caso**, si existen múltiples colisiones mal gestionadas, el tiempo se degrada a:

$$\boxed{T(n) = \mathcal{O}(n^2)}$$

Por lo tanto, el rendimiento de Hash Sort depende fuertemente del diseño de la función hash y del control del factor de carga en la tabla hash.