**DSA Sorting MCQ**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

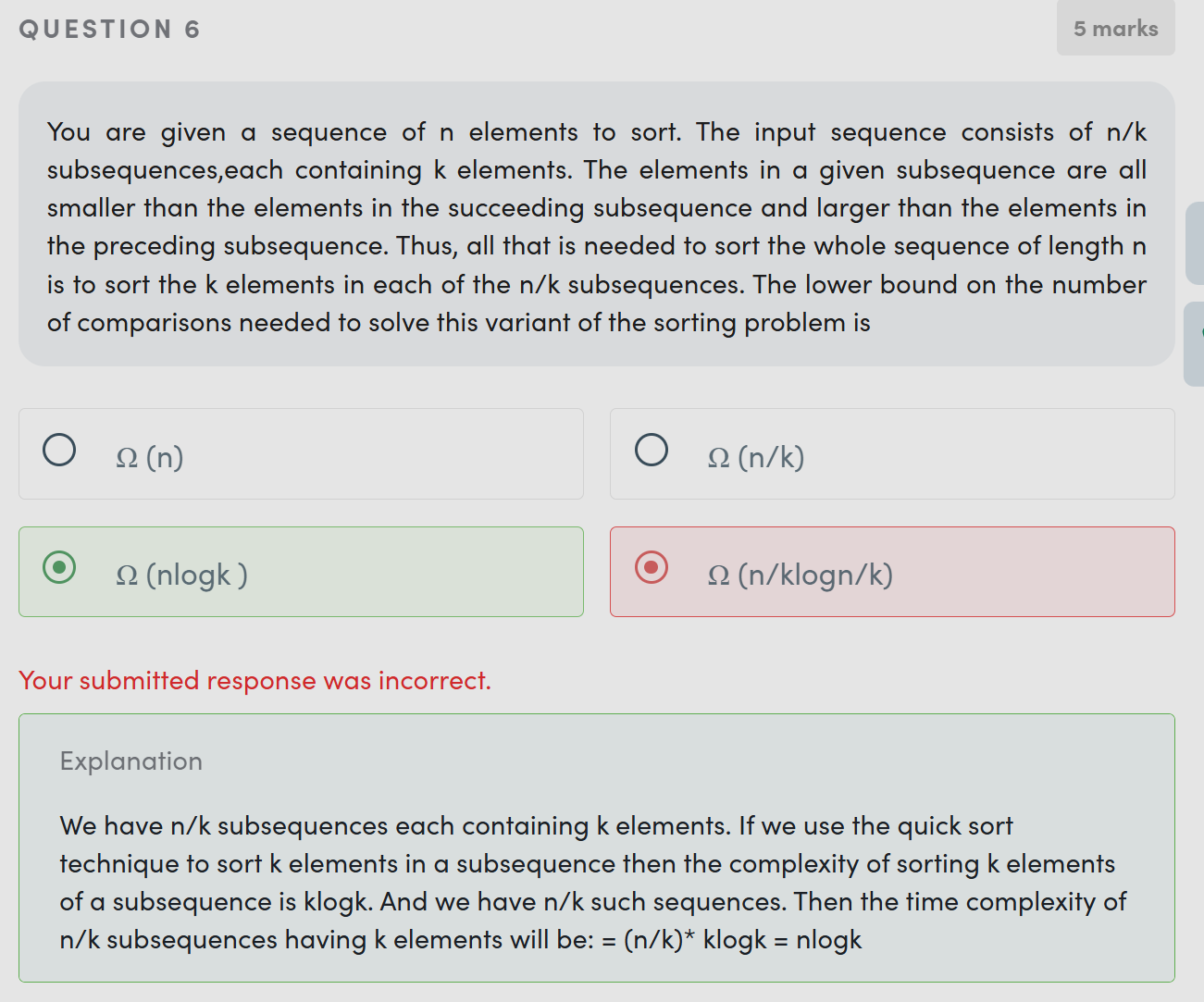
AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

**A screenshot of a survey

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a test

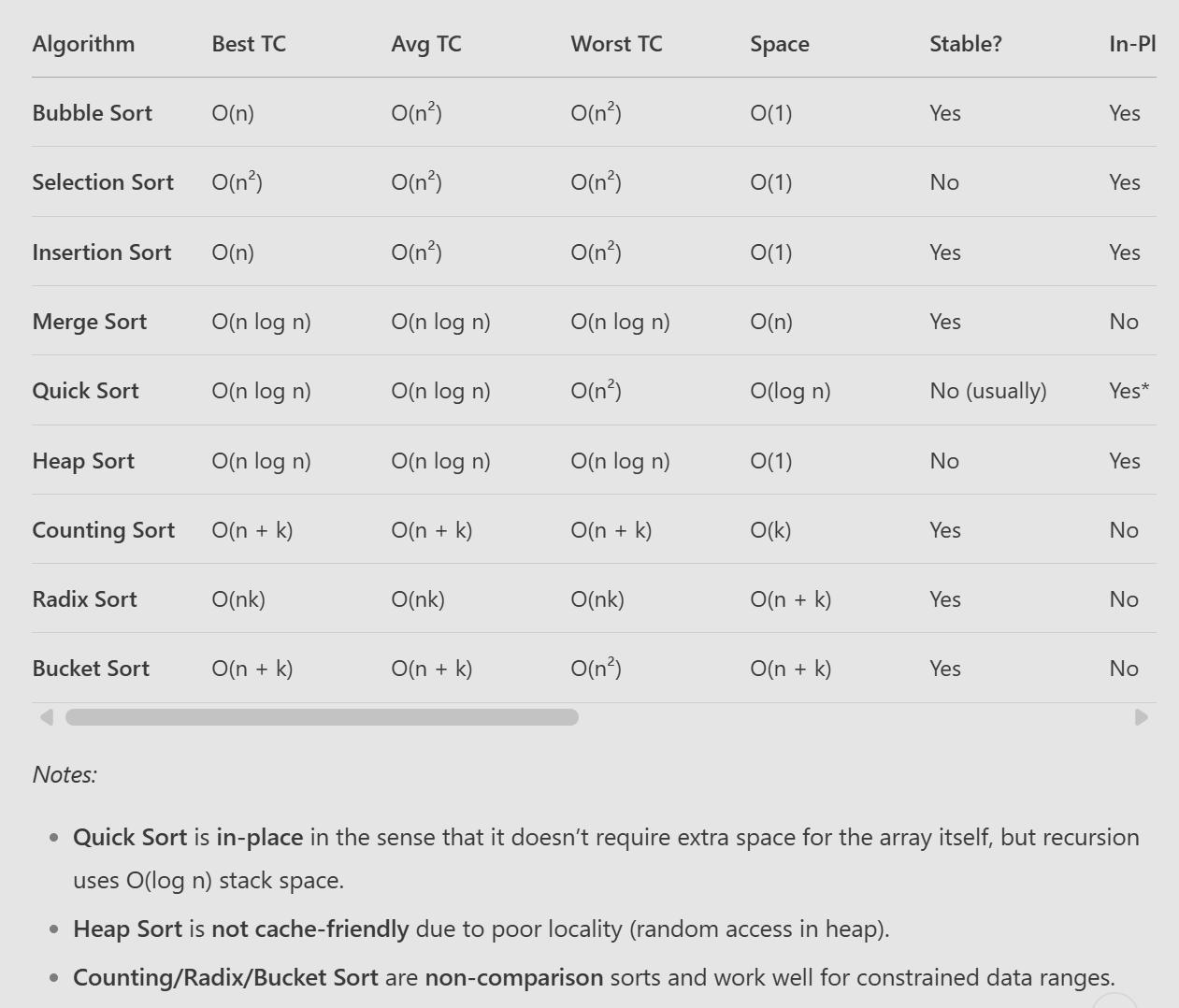
AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

**3. Key Comparisons & When to Use Which?**

| **Scenario** | **Best Algorithm(s)** | **Reason** |
| --- | --- | --- |
| **Small datasets (n < 100)** | Insertion Sort | Low overhead, adaptive, stable |
| **Nearly sorted data** | Insertion Sort, Bubble Sort | Adaptive, O(n) best case |
| **Large datasets (general case)** | Quick Sort (avg case), Merge Sort (worst case) | Fast avg-case, cache-friendly (QuickSort) |
| **Worst-case O(n log n) needed** | Heap Sort, Merge Sort | No O(n²) risk |
| **External sorting (disk/tape)** | Merge Sort | Sequential access, handles large data |
| **Fixed small range (e.g., 0-100)** | Counting Sort | O(n + k) time, linear |
| **Large numbers with digits/strings** | Radix Sort | O(nk) time, efficient for digits |
| **Uniformly distributed floats** | Bucket Sort | O(n) avg if well-distributed |
| **Minimizing writes (e.g., flash memory)** | Selection Sort | Only O(n) swaps |

**A white background with black text

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A close-up of a computer code

AI-generated content may be incorrect.**