Change log:

- Problem 2 (version 3): specified that the partition is on edges: the pieces may share vertices.
- Problem 3&4 (version 3): added condition that $A[i] > i$, and distinguished starting and ending points.
- Problems 9 & 10 (version 2): changed to finding a TSP walk (finish at different vertex than starting), instead of tour (finish and start at same vertex).

This problem set has a total of 10 written problems, plus 2 coding problems, on 2 pages. Written solutions should be submitted to Canvas by 2pm Wednesday Apr 21, 2021. Coding solutions should be submitted separately.

1. Consider a perfect $d$-level $B$ tree: each node has $B$ children, with leaves numbered 1 $\ldots$ $B^d$. Express the number of $B$ tree nodes accessed when querying the range $[l, r]$, in terms of the base-B representation of $l$ and $r$.

   Recall a node is accessed if its corresponding list of values is contained in the range, but its parent’s list is not.

2. Show that any tree on $n$ vertices, for any parameter $B$, can be partitioned, in $\tilde{O}(n)$ time, into $O(n/B)$ edge-disjoint pieces (that may overlap at vertices) such that:
   
   (a) The number of edges contained in each piece is at most $B$
   (b) Each piece has at most 2 vertices with neighbors to other pieces.

3. Consider a sequence specifying ‘jump’ pointers: $A[i]$ says that if we are at $i$, we can get to location $A[i] > i$ in one step. Note that these pointers only go rightward. Show that a sequence of:
   
   (a) $\text{MODIFY}(i, x)$ modify some $A[i]$ to $x > i$,
   (b) $\text{QUERY}(l, r)$: return how many ‘jumps’ it takes to get past location $r$ if we start at $l$,

   can be answered in $O(n^{1.9})$ time total.

4. Same as above, but in $O(n^{1.1})$ time.
5. The number of local maximas of a sequence \( A[l \ldots r] \) to be the number of indices \( l \leq i \leq r \) such that
\[
A[i] > \min_{l \leq j < i} A[j],
\]
and the range local maxima query on an array \( A \) asks for the number of local maximas in the range \([l, r]\). Show that a sequence of \( n \) modifies and range local maxima queries on a sequence of length \( n \) can be answered in \( O(n^{1.9}) \) time.

6. Same as above, but in \( O(n^{1.1}) \) time.

7. Define the distinct sum of a list of numbers to be the sum of the set of numbers with duplicates removed. The range max interval distinct sum query asks, for a given \([l, r]\), the maximum distinct sum of some contained interval \([\hat{l}, \hat{r}]\) with \( \hat{l} \geq l \) and \( \hat{r} \leq r \). Show that \( n \) range max interval distinct sum queries on a static array can be answered in \( O(n^{1.9}) \) time.

8. Same as above, but in \( O(n^{1.1}) \) time.

9. Given a undirected graph on \( n \) vertices consisting of \( n - 1 \) edge with weight 1, and the rest containing edges of weight at least \( \lceil n/3 \rceil \), show that the TSP walk (min weighted walk that visits each vertex at least once) uses a non-tree edge at most once. Note that this walk does not need to end where it started.

10. Give an \( \tilde{O}(n^{1.1}) \) time or faster algorithm for finding, in a weighted undirected graph with a tree and some undirected edges, the min weight walk that uses an off-tree edge at most once.