**Longest Increasing Subsequence**

**Code:**

**package** Subsequence;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** Longest {

**public** **static** List<Integer> findLongest(**int**[] nums) {

**int** n = nums.length;

**int**[] lengths = **new** **int**[n];

**int**[] previousIndices = **new** **int**[n];

Arrays.*fill*(previousIndices, -1);

**int** maxLength = 0;

**int** endIndex = 0;

**for** (**int** i = 0; i < n; i++) {

lengths[i] = 1;

**for** (**int** j = 0; j < i; j++) {

**if** (nums[j] < nums[i] && lengths[j] + 1 > lengths[i]) {

lengths[i] = lengths[j] + 1;

previousIndices[i] = j;

}

}

**if** (lengths[i] > maxLength) {

maxLength = lengths[i];

endIndex = i;

}

}

List<Integer> longest = **new** ArrayList<>();

**while** (endIndex >= 0) {

longest.add(nums[endIndex]);

endIndex = previousIndices[endIndex];

}

**int** sequenceLength = longest.size();

**for** (**int** i = 0; i < sequenceLength / 2; i++) {

**int** temp = longest.get(i);

longest.set(i, longest.get(sequenceLength - i - 1));

longest.set(sequenceLength - i - 1, temp);

}

**return** longest;

}

**public** **static** **void** main(String[] args) {

**int**[] nums = {2, 3, 4, 1, 10};

List<Integer> longest = *findLongest*(nums);

System.***out***.println("Longest increasing subsequence: " + longest);

}

}