

---

```

1 // O(n)
2 void k_closest_to_median(vector<int>& arr, int k){
3     double median = median_selection(arr); // O(n)
4     vector<my_pair> diff;
5     // O(n)
6     for(int e : arr){
7         diff.push_back({abs(e - median), e});
8     }
9     // O(n)
10    int kth_idx = quick_select_idx(diff, k);
11    //O(n)
12    for(int i = kth_idx; i >= 0; i--){
13        cout << diff[i].y << " ";
14    }
15    cout << endl;
16 }

```

---

Line 3 of the algorithm uses quickselect to find the median and is on average  $O(n)$ . Lines 6 – 8 will create a diff array that holds the value and its absolute difference from the median of the original array. This is  $\Theta(n)$  time. Line 10 uses quickselect to find the  $k^{th}$  least element, this is on average  $O(n)$ . lines 12-14 print out the  $k$  closest elements to the median, this is  $\Theta(k)$  where  $k$  is at most  $n$ . In total the algorithm runs on average  $O(n)$ .