Activity Selection Problem Experiment

6. Activity Selection Problem (Greedy Method)

C Code

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int start;
    int finish;
} Activity;

int compareActivities(const void *a, const void *b) {
    Activity *actA = (Activity *)a;
    Activity *actB = (Activity *)b;

    // Patched: Was actA->finish - actA->finish, which is always 0.
    return actA->finish - actB->finish;
}

void solve_activity_selection(Activity activities[], int n) {
    qsort(activities, n, sizeof(Activity), compareActivities);

    int last_finish_time = activities[0].finish;
    printf("(%d, %d)\n", activities[0].start, activities[0].finish);

    for (int i = 1; i < n; i++) {
        if (activities[i].start >= last_finish_time) {
            printf("(%d, %d)\n", activities[i].start, activities[i].finish);
            last_finish_time = activities[i].finish;
        }
    }
}

int main() {
    Activity jobs[] = {
        {1, 4}, {3, 5}, {0, 6}, {5, 7}, {3, 8}, {5, 9},
        {6, 10}, {8, 11}, {8, 12}, {2, 13}, {12, 14}
    };
    int n1 = sizeof(jobs) / sizeof(jobs[0]);
    solve_activity_selection(jobs, n1);

    printf("---\n"); // Minimal separator

    Activity jobs_2[] = {
        {1, 3}, {2, 4}, {3, 5}, {0, 6}, {5, 7}, {8, 9}, {5, 9}
    };
    int n2 = sizeof(jobs_2) / sizeof(jobs_2[0]);
    solve_activity_selection(jobs_2, n2);

    return 0;
}
```