

به نام خدا



درس: برنامه‌نویسی و ساختارهای موازی

استاد: دکتر هاشمی

گزارش پروژه نهایی درس

سیدمحمدامین منصوری طهرانی

۹۴۱۰۵۱۷۴

لازم به ذکر است که بنده و آقای حمیدرضا احمدی به اتفاق هم این پژوهش را در این ایام تکمیل کردیم و تشابه میان فایل ها نیز به همین دلیل است.

در ابتدا به ترتیب به توضیح بخش های مختلف کد می پردازم و در انتهای نمونه هایی از خروجی ها را می آورم.

در تصویر زیر کتابخانه های لازم *include* شده و از جمله ای آنها کتابخانه های مربوط به *thrust* هستند. بعد از آن هم *prototype* ها نوشته شده اند

```

1 //DO NOT MODIFY THIS FILE
2
3 // #include "knn.h"
4 #include <stdio.h>
5 #include <time.h>
6 #include <stdlib.h>
7 #include <math.h>
8 // #include <thrust/reduce.h>
9 #include <thrust/device_ptr.h>
10 #include <thrust/sort.h>
11 #include "cuda_runtime.h"
12 #include "device_launch_parameters.h"
13 #include <thrust/extrema.h>
14
15 #include "gputimer.h"
16 #include "gpuerrors.h"
17
18 #define D 128
19 #define D_L 100
20 #define N_Ref 1000000
21
22 // =====> Functions Prototype <=====
```

- 23 int fvecs_read (const char *fname, int d, int n, float *a);
- 24 int ivecs_read (const char *fname, int d, int n, int *a);
- 25 int ivecs_write (const char *fname, int d, int n, const int *v);
- 26 int call_error(int* nn_true, int* nn_predict, unsigned int N, unsigned int K);
- 27 void get_inputs(int argc, char *argv[], unsigned int& N, unsigned int& K);
- 28 void gpuKernels(float* ref, float* query, int* pred, unsigned int N, unsigned int K, double* gpu_kernel_time, int* sort_val);
- 29 // void gpuKernels(float* ref, float* query, int* pred, float* a, unsigned int N, unsigned int K, double* gpu_kernel_time);

در تصویر زیر تابع کرنل اول برای محاسبه فاصله های اقلیدسی نوشته شده است. ورودی اول آن کل بردارهای مرجع بوده که یک میلیون بردار با ۱۲۸ بعد هستند.

```

32 // =====
33 __global__ void kernelFunc1(float* d_ref, float* d_q , float* d_dist) {
34     __shared__ float sum_arr[128];
35     int block, index;
36     long long i;
37     i = blockIdx.y * gridDim.x + blockIdx.x;    //y * 1000 + x
38     block = i;
39     i *= blockDim.x * blockDim.y * blockDim.z;   //128 * 1 * 1
40     i += threadIdx.x;
41     index = threadIdx.x;
42
43     sum_arr[index] = pow(* (d_ref + i) - *(d_q + index), 2);
44
45     __syncthreads();
46
47     int counter = 128;
48     while(counter >= 2){
49         if(index < counter/2){
50             sum_arr[index] += sum_arr[index + counter/2];
51             __syncthreads();
52         }
53         counter = counter/2;
54     }
55     d_dist[block] = sqrt(sum_arr[0]);
56 }
```

روش محاسبه:

با توجه به وجود یک میلیون بردار مرجع، یک ماتریس ۱۰۰۰ در ۱۰۰۰ در نظر می گیرم که هر درایه این ماتریس نشان دهنده یک *thread* می باشد. در هر *thread block* تعداد *thread* ۱۲۸ اجرا می شوند و وظیفه هر کدام محاسبه مجدد

اختلاف دو عدد در یک بعد از ۱۲۸ بعد مربوط به بردارها هستند. برای مثال اگر ترد ۱۰۰ ام که در $bx = 1$ و $by = 1$ است را در نظر بگیریم، محاسبه‌های این *thread block* برای بدست آوردن فاصله اقلیدسی داده *query* داده شده به کرنل با بردار مرجع ۱۰۰۱ ام هستند. هر ترد مجبور اختلاف المان متناظر اندیس آن ترد در نقطه *query* را با المان متناظر اندیس آن ترد در بردار مرجع ۱۰۰۱ ام حساب می‌کند. با سنتکرون کردن آن‌ها همه عملیات مورد نظر را انجام می‌دهند و مشکلات *read before write* و از این دست به وجود نمی‌آید. پس از آن با توجه به این‌که این مجبور اختلاف‌ها برای افزایش سرعت و بهبود *shared memory performance* در ذخیره شده‌اند، برای محاسبه فاصله نقطه ۱۲۸ بعدی نقطه ۱۲۸ با بردار مرجع ۱۰۰۱ ام، عملیات جمع این ۱۲۸ به صورت موازی یا همان *parallel reduce* به صورت *reduce* انجام می‌شود. آرگومان سوم کرنل نیز آرایه‌ای است که فاصله همه بردارهای مرجع را با یک نقطه *query* نگه می‌دارد و با اجرای همه تردهای یک *thread block*، پس از عملیات *reduce* مقداری که در خانه اندیس صفر *sum_arr* قرار دارد همان مجبور فاصله مورد نظر است که از آن جذر گرفته و به عنوان فاصله باز می‌گردانیم. به این ترتیب این کرنل تمام می‌شود و با هر بار اجرا شدن آن، فاصله همه یک میلیون بردار مرجع تا نقطه *query* که به آن داده می‌شود بدست می‌آید.

در این تصویر دقیقاً همان عملیات بالا انجام می‌شود و تنها تفاوت این است که فاصله منهتن سنجیده می‌شود.

```

58  global__ void kernelFunc2(float* d_ref, float* d_q , float* d_dist) {
59  __shared__ float sum_arr[128];
60  int block, index;
61  long long i;
62  i = blockDim.y * gridDim.x + blockDim.x; //y * 1000 + x
63  block = i;
64  i *= blockDim.x * blockDim.y * blockDim.z; //128 * 1 * 1
65  i += threadIdx.x;
66  index = threadIdx.x;
67
68  sum_arr[index] = abs(*(d_ref + i) - *(d_q + index));
69
70  __syncthreads();
71
72  int counter = 128;
73  while(counter >= 2){
74      if(index < counter/2){
75          sum_arr[index] += sum_arr[index + counter/2];
76          __syncthreads();
77      }
78      counter = counter/2;
79  }
80  // sum[block] = thrust::reduce(sum_arr, sum_arr + 128, 0, thrust::plus<int>());
81  d_dist[block] = (sum_arr[0]);
82 }
```

در تصویر نیز مانند دو کرنل قبلی نوشته شده است با این تفاوت اندک که برای محاسبه فاصله نقطه *query* با بردارهای مرجع باید ۳ مقدار را برای هر جفت المان در یک بعد یکسان نگه داریم که منجر به استفاده از ۳ آرایه در *shared memory* می‌شود. در ادامه هر سه آرایه به صورت موازی *reduce* شده و فاصله منهتن به عنوان خروجی در آرایه فاصله‌ها نوشته می‌شود.

```

84  __global__ void kernelFunc3(float* d_ref, float* d_q , float* d_dist) {
85      __shared__ float mult[128];
86      __shared__ float a_square[128];
87      __shared__ float b_square[128];
88
89      int block, index;
90      long long i;
91      i = blockIdx.y * gridDim.x + blockIdx.x;    //y * 1000 + x
92      block = i;
93      i *= blockDim.x * blockDim.y * blockDim.z;   //128 * 1 * 1
94      i += threadIdx.x;
95      index = threadIdx.x;
96
97      a_square[index] = pow(*(d_ref + i), 2);
98      b_square[index] = pow(*(d_q + index), 2);
99      mult[index] = (*(d_ref + i)) * (*(d_q + index));
100
101     __syncthreads();
102
103     int counter = 128;
104     while(counter >= 2){
105         if(index < counter/2){
106             a_square[index] += a_square[index + counter/2];
107             b_square[index] += b_square[index + counter/2];
108             mult[index] += mult[index + counter/2];
109             __syncthreads();
110         }
111         counter = counter/2;
112     }
113     d_dist[block] = mult[0]/(sqrt(a_square[0] * b_square[0]));
114 }
```

*** تصویر زیر قطعه کد کرنلی است که در اجرای کد از آن استفاده شده است. چون در هنگام مرتب کردن به آرایه‌ای از اندیس‌ها نیاز داشتم کرنلی تعریف کرم که با اجرا شدن آرایه‌ای را روی *device* به صورت مرتب با اعداد صفر تا ۹۹۹۹۹ پر کند و سریع این کار انجام شود.

```

116  __global__ void gpu_value_init(int* array){
117      int block;
118      long long i;
119      i = blockIdx.y * gridDim.x + blockIdx.x;    //y * 1000 + x
120      block = i;
121      array[block] = block;
122 }
```

تصویر زیر قسمتی از تابع *main* را نشان می‌دهد که من دو متغیر جدید تعریف کرده‌ام که اولی *sort_value* بوده و در این آرایه اعداد ۰ تا ۹۹۹۹ به ترتیب ذخیره می‌شوند و به *device* کپی خواهند شد تا توسط کرنلی بر روی *gpu* به عنوان *key* برای مرتب کردن اندیس فاصله‌هایی که کمترین فاصله را دارند استفاده شوند. متغیر *integer* که تعریف کردم (index) نیز برای چاپ خروجی‌ها و نمایش آن‌ها در گزارش بود و تأثیر دیگری ندارد.

```

124 int main(int argc, char *argv[]) {
125
126     struct cudaDeviceProp p;
127     cudaGetDeviceProperties(&p, 0);
128     printf("Device Name: %s\n", p.name);
129
130     // get parameters from command line
131     unsigned int N, K;
132     get_inputs(argc, argv, N, K);
133
134     // allocate memory in CPU for calculation
135     float* reference; // reference vectors
136     float* query; // query points
137     int* nn_true; // groundtruth for the query points
138     int* nn_predict;
139     // float* distances;
140     int* sort_value;
141     int index;
142
143     // Memory Allocation
144     reference = (float*)malloc(N_ref * D * sizeof(float));
145     query = (float*)malloc(N * D * sizeof(float));
146     nn_true = (int*)malloc(D_L * N * sizeof(int));
147     nn_predict = (int*)malloc(3 * K * N * sizeof(int));
148     // distances = (float*)malloc(N_ref * sizeof(float));
149     sort_value = (int*)malloc(N_ref * sizeof(int));
150     // fill references, query and labels with the values read from files
151     fvecs_read("/home/data/ref.fvecs", D, N_ref, reference);
152     fvecs_read("/home/data/query.fvecs", D, N, query);
153     ivec3s_read("/home/data/groundtruth.ivecs", D_L, N, nn_true);
154     printf("ground truths read\n");

```

در تصویر زیر همان عملیات چاپ کردن آورده شده است. خروجی *nn_predict* به این صورت است: به تعداد $N \times K$ عدد اول به فاصله اقلیدسی مربوط بوده و $N \times K$ داده بعدی به فاصله منهتن و فاصله های بعدی نیز برای روش کسینوسی هستند. طبیعتاً داده اول، K همسایه نزدیک با معیار اقلیدسی به داده *query* اول بوده و K داده دوم K همسایه نزدیک با همین معیار به داده دوم بوده و این روند ادامه دارد تا $N \times K$ داده اول تمام شوند و پس از آن به همین ترتیب فاصله های منهتن و کسینوسی می آیند.

```

156     for(int j=0; j<N_ref; j++){
157         sort_value[j] = j;
158     }
159     // time measurement for GPU calculation
160     double gpu_kernel_time = 0.0;
161     clock_t t0 = clock();
162     gpuKernels(reference, query, nn_predict, N, K, &gpu_kernel_time, sort_value);
163     for(int i=0; i< 3 * N * K; i++){
164         index = ((i/K) * D_L) + (i % K);
165         if(i < N * K){
166             printf("Euclidean\t N is %d, gth is: %d \t and prediction is: %d\n", i % K, nn_true[index], nn_predict[i]);
167             if ( i % K == 0 ){
168                 printf("\n\nQuery num = %d", (i/K) % 3);
169                 printf("\n\n\n");
170             }
171         }
172         else if(i < 2 * N * K){
173             printf("Manhattan\t N is %d, gth is: NULL \t and prediction is: %d\n", i % K, nn_predict[i]);
174             if ( i % K == 0 ){
175                 printf("\n\nQuery num = %d", (i/K) % 3);
176                 printf("\n\n\n");
177             }
178         }
179         else{
180             printf("Cosine\t N is %d, gth is: NULL \t and prediction is: %d\n", i % K, nn_predict[i]);
181             if ( i % K == 0 ){
182                 printf("\n\nQuery num = %d", (i/K) % 3);
183                 printf("\n\n\n");
184             }
185         }
186     }

```

در تصویر زیر هم اتفاق خاصی به جز چاپ نتیجه نمی افتد!تابع محاسبه خطرا در انتهای توضیح داده ام. فایل خروجی نیز اینجا نوشته می شود.

```

188     clock_t t1 = clock();
189
190     // check correctness of calculation
191     double acc = 1 - (double)calc_error(nn_true, nn_predict, N, K) / N*K;
192     printf("k=%d n=%d GPU=%g ms GPU-Kernels=%g ms accuracy=%f\n",
193           K, N, (t1-t0)/1000.0, gpu_kernel_time, acc);
194
195     // write the output to a file
196     ivec3s_write("output.ivec3s", K, N, nn_predict);
197
198     // free allocated memory for later use
199     free(reference);
200     free(nn_predict);
201     free(nn_true);
202     free(query);
203
204     return 0;
205 }
```

در تصویر زیر تابعی که برای فرآخوانی کرنل‌ها بر روی *gpu* نوشته شده است دیده می‌شود. برای نگه داشتن بردارهای مرجع بر روی *d_query* و *d_device* نیز به همین منظور تعریف می‌شود. فاصله‌های *query* را با *D_distance* و *D_reference* معرفی می‌کنند. در آخر هم *block* و *grid* مطابق روشی که بالاتر توضیح داده شد تعریف شده‌اند.

```

208 void gpuKernels(float* reference, float* query, int* nn_predict, unsigned int N, unsigned int K, double* gpu_kernel_time, int* sort_val) {
209     // void gpuKernels(float* reference, float* query, int* nn_predict, float* d, unsigned int N, unsigned int K, double* gpu_kernel_time) {
210
211     // Memory Allocation and Copy to Device
212
213     GpuTimer timer;
214     timer.Start();
215     // printf("\nHello1");
216     float* d_reference;
217     float* d_query;
218     // int* d_nn_predict;
219     float* d_distances;
220     // int* d_value;
221     int* d_sort_val;
222
223     HANDLE_ERROR(cudaMalloc((void**)&d_reference, N_ref * D * sizeof(float)));
224     HANDLE_ERROR(cudaMalloc((void**)&d_query, N * D * sizeof(float)));
225     // HANDLE_ERROR(cudaMalloc((void**)&d_nn_predict, 3 * D_L * N * sizeof(int)));
226     HANDLE_ERROR(cudaMalloc((void**)&d_distances, N_ref * sizeof(float)));
227     HANDLE_ERROR(cudaMalloc((void**)&d_value, N_ref * sizeof(int)));
228     HANDLE_ERROR(cudaMalloc((void**)&d_sort_val, N_ref * sizeof(int)));
229
230     HANDLE_ERROR(cudaMemcpy(d_reference, reference, N_ref * D * sizeof(float), cudaMemcpyHostToDevice));
231     HANDLE_ERROR(cudaMemcpy(d_query, query, N * D * sizeof(float), cudaMemcpyHostToDevice));
232     HANDLE_ERROR(cudaMemcpy(d_sort_val, sort_val, N_ref * sizeof(int), cudaMemcpyHostToDevice));
233
234     // dim3 grid ( 512 * 128, 1, 1 ); // This is not correct because 512 * 128 65536 > 65535
235     dim3 grid (1000, 1000, 1); //x,y,z
236     dim3 block (128, 1, 1); //x,y,z
237
238 }
```

در تصویر زیر سه حلقه وجود دارد که هر کدام فاصله‌ها را برای یک روش محاسبه کرده و در *nn_predict* به نحوی که توضیح داده شد کپی می‌کنند. یک نکته قابل توجه تغییر کردن آدرس ابتدای *query* است که به کرنل محاسبه کننده فاصله داده می‌شود. در واقع حلقه به تعداد نقاط *query* اجرا می‌شود پس باشد هر بار ابتدای آرایه‌ای که به کرنل داده می‌شود ۱۲۸ عدد جلو برود تا به نقطه *query* بعدی برسیم.

```

240     for(int m=0; m<N; m++){
241         HANDLE_ERROR(cudaMemcpy(d_sort_val, sort_val, N_ref * sizeof(int), cudaMemcpyHostToDevice));
242         kernelFunc1<<< grid , block>>>(d_reference, &(d_query[m * 128]), d_distances);
243
244         thrust::device_ptr<float> t_dist(d_distances);
245         thrust::device_ptr<int> t_value(d_sort_val);
246         thrust::sort_by_key(t.dist, t.dist + N_ref, t_value);
247         HANDLE_ERROR(cudaMemcpy(&nn_predict[m * K]), d_sort_val, K * sizeof(int), cudaMemcpyDeviceToHost));
248     }
249
250     for(int m=0; m<N; m++){
251         HANDLE_ERROR(cudaMemcpy(d_sort_val, sort_val, N_ref * sizeof(int), cudaMemcpyHostToDevice));
252         kernelFunc2<<< grid , block>>>(d_reference, &(d_query[m * 128]), d_distances);
253
254         thrust::device_ptr<float> t_dist(d_distances);
255         thrust::device_ptr<int> t_value(d_sort_val);
256
257         thrust::sort_by_key(t.dist, t.dist + N_ref, t_value);
258         HANDLE_ERROR(cudaMemcpy(&nn_predict[(m + N) * K]), d_sort_val, K * sizeof(int), cudaMemcpyDeviceToHost));
259     }
260
261     for(int m=0; m<N; m++){
262         HANDLE_ERROR(cudaMemcpy(d_sort_val, sort_val, N_ref * sizeof(int), cudaMemcpyHostToDevice));
263         kernelFunc3<<< grid , block>>>(d_reference, &(d_query[m * 128]), d_distances);
264
265         thrust::device_ptr<float> t_dist(d_distances);
266         thrust::device_ptr<int> t_value(d_sort_val);
267
268         thrust::sort_by_key(t.dist, t.dist + N_ref, t_value);
269         HANDLE_ERROR(cudaMemcpy(&nn_predict[(m + 2*N) * K]), d_sort_val, K * sizeof(int), cudaMemcpyDeviceToHost));
270         // HANDLE_ERROR(cudaMemcpy(&nn_predict[(m + 2*N) * K]), &(d_sort_val[N_ref - K - 1]), K * sizeof(int), cudaMemcpyDeviceToHost));
271     }
272 }

```

عملیات اصلی دیگری که در هر لود حلقه انجام می‌شود مرتب کردن فاصله‌هاست. این کار به وسیله کتابخانه بسیار *efficient* موجود برای *cuda* به نام *thrust* انجام می‌شود. مرتب کردن به وسیله این کرنل بر روی *gpu* انجام می‌شود و برای این کار به آرایه فاصله‌ها بر روی *key* و آرایه *device* ها برای بدست آوردن اندیس (آن هم بر روی *device*) نیاز است. برای استفاده از آن‌ها در *thrust* باید آن‌ها را مطابق توضیحات *documentation* به *smart pointer* کنیم. این کرنل ابتدای بازه و انتهای بازه‌ای که از آرایه که قصد مرتب کردنش را داریم می‌گیرد و به صورت درجا در خود آرایه اعداد را مرتب می‌کند و به این ترتیب اندیس‌ها در *d_sort_val* به صورت مرتب قرار خواهند داشت. برای کپی در *host* فقط *K* تای اول آن را کپی می‌کنیم. (تنظیم آرگومان سوم *cudaMemcpy* همین روند برای دو معیار فاصله دیگر برای همه نقاط *query* انجام می‌شود و این قسمت به پایان می‌رسد).

در تصویر زیر همه آرایه‌هایی که بر روی *device* تعریف شده بودند رها (*Free*) می‌شوند. تابع *get_inputs* هم برای خواندن *N* و *K* بوده که توسط من نوشته نشده و توضیح ندارد ():

```

275     cudaFree(d_reference);
276     cudaFree(d_query);
277     // cudaFree(d_nn_predict);
278     cudaFree(d_distances);
279     cudaFree(d_value);
280     cudaFree(d_sort_val);
281     timer.Stop();
282     *gpu_kernel_time = timer.Elapsed();
283
284     //Copy to Host and Free the Memory
285
286 }
287 //-----
288 void get_inputs(int argc, char *argv[], unsigned int& N, unsigned int& K)
289 {
290     if (
291         argc != 3 ||
292         atoi(argv[1]) < 0 || atoi(argv[1]) > 10000 ||
293         atoi(argv[2]) < 0 || atoi(argv[2]) > 100
294     ) {
295         printf("<< Error >>\n");
296         printf("Enter the following command:\n");
297         printf("\t./knn N K\n");
298         printf("\t\tN must be between 0 and 10000\n");
299         printf("\t\tK must be between 0 and 100\n");
300         exit(-1);
301     }
302     N = atoi(argv[1]);
303     K = atoi(argv[2]);
304 }

```

در تصویر آخر محاسبه خطأ به وسیله تابع *calc_error* انجام می‌شود. از آن جایی که $3 \times K \times N$ اندیس به عنوان *nn_true* در آرایه *nn_predict* ذخیره شده‌اند و *groundtruth prediction* فقط برای روش اقلیدسی موجود است، باید فقط از داده اول برای مقایسه با *nn_true* استفاده کنیم. در این قسمت همین کار انجام شده و تعداد خطاهای کل $N \times K$ می‌شود و به عنوان خروجی بازگردانده می‌شود.

```

404 int calc_error(int* nn_true, int* nn_predict, unsigned int N, unsigned int K) {
405     // int fault = 0;
406     int sum = 0;
407     int index;
408     for(int n=0; n< N * K; n++){
409         index = ((n/K) * D_L) + (n % K);
410         // printf("\n%d\n", index);
411         if(nn_predict[n] != nn_true[index]){
412             sum += 1;
413         }
414     }
415     return sum/(N * K);
416 }
```

در ادامه خروجی‌ها را به ازای چند حالت مختلف نشان می‌دهیم. دقت کنید که برای دو روش دیگر فقط *prediction* مقدار *ground truth* را برای آن‌ها *NULL* چاپ می‌کنیم.

توجه: دو ساعت مانده به ددلاین ایده قسمت ** به ذهنم رسید ولی به دلیل اشغال بودن ۶ گیگ از حافظه سرور توسط کاربری دیگر امکان اجرا کردن کد را نداشتیم. اما اگر امکان داشت لطفاً خطوط ۲۴۱ و ۲۵۱ و ۲۶۴ و ۲۵۲ را کامنت کنید و خطوط ۲۴۳ و ۲۵۲ و ۲۶۶ را *uncomment* کنید. حدس من این است که به علت رفع شدن *overhead* ناشی از *cudaMemcpy* کمتر شود. با تشکر از $N = 10000, K = 100$ ها سرعت اجرا شدن حتی از ۳ دقیقه هم برای

توجه شما.

خروجی‌های زیر در حالتی هستند که اتفاق بالا نیفتاده است.

اجرای کد تنها بدون چاپ هیچ خروجی - دقت ۱ بوده و زمان اجرا ۲۱۰ ثانیه.

```

Last login: Tue Jan 29 07:08:57 2019 from 213.233.188.138
s107@sumi:~$ cd the/
s107@sumi:~/the$ cd project/
s107@sumi:~/the/project$ sl
The program 'sl' is currently not installed. To run 'sl' please ask your administrator to install the package 'sl'
s107@sumi:~/the/project$ ls
Desktop      gputimer.h  knn_94105174.ivecs  knn.cu          unsafe2.cu
gpuerrors.h  knn        knn_94105174.txt   last_without_8.cu  unsafe.cu
s107@sumi:~/the/project$ ./knn 10000 100
Device Name: GeForce GTX 1080
ground truths read
k=100 n=10000 GPU=210483 ms GPU-Kernels=210712 ms accuracy=1.000000
s107@sumi:~/the/project$
```

اجرای کد با چاپ K همسایه‌های نزدیک برای تعداد نقاط *query* اندک (که بتوان مشاهده کرد).
اگر K برابر ۱ باشد با ۱۰ نقطه *:query*

```
s107@sumi:~/the/project$ ./knn 10 1
Device Name: GeForce GTX 1080
ground truths read
Euclidean      N is 0, gth is: 932085      and prediction is: 932085

Query num = 0

Euclidean      N is 0, gth is: 413247      and prediction is: 413247

Query num = 1

Euclidean      N is 0, gth is: 669835      and prediction is: 669835

Query num = 2

Euclidean      N is 0, gth is: 970797      and prediction is: 970797

Query num = 0

Euclidean      N is 0, gth is: 340871      and prediction is: 340871

Query num = 1

Euclidean      N is 0, gth is: 187470      and prediction is: 187470

Query num = 2

Euclidean      N is 0, gth is: 402219      and prediction is: 402219

Query num = 0

Euclidean      N is 0, gth is: 906750      and prediction is: 906750

Query num = 1

Euclidean      N is 0, gth is: 376328      and prediction is: 376328

Query num = 2

Euclidean      N is 0, gth is: 178811      and prediction is: 178811

Query num = 0
```

```
Query num = 0

Manhattan      N is 0, gth is: NULL      and prediction is: 934876

Query num = 1

Manhattan      N is 0, gth is: NULL      and prediction is: 413247

Query num = 2

Manhattan      N is 0, gth is: NULL      and prediction is: 408764

Query num = 0

Manhattan      N is 0, gth is: NULL      and prediction is: 889039

Query num = 1

Manhattan      N is 0, gth is: NULL      and prediction is: 175336

Query num = 2

Manhattan      N is 0, gth is: NULL      and prediction is: 187470

Query num = 0

Manhattan      N is 0, gth is: NULL      and prediction is: 982379

Query num = 1

Manhattan      N is 0, gth is: NULL      and prediction is: 906750

Query num = 2

Manhattan      N is 0, gth is: NULL      and prediction is: 376328

Query num = 0

Manhattan      N is 0, gth is: NULL      and prediction is: 177646

Query num = 1
```

```
Cosine  N is 0, gth is: NULL    and prediction is: 423462
Query num = 2

Cosine  N is 0, gth is: NULL    and prediction is: 217625
Query num = 0

Cosine  N is 0, gth is: NULL    and prediction is: 677133
Query num = 1

Cosine  N is 0, gth is: NULL    and prediction is: 953479
Query num = 2

Cosine  N is 0, gth is: NULL    and prediction is: 824142
Query num = 0

Cosine  N is 0, gth is: NULL    and prediction is: 443412
Query num = 1

Cosine  N is 0, gth is: NULL    and prediction is: 814846
Query num = 2

Cosine  N is 0, gth is: NULL    and prediction is: 443597
Query num = 0

Cosine  N is 0, gth is: NULL    and prediction is: 986614
Query num = 1

Cosine  N is 0, gth is: NULL    and prediction is: 776353
Query num = 2

k=1 n=10 GPU=2095.23 ms GPU-Kernels=1848.17 ms accuracy=1.000000
s107@sumi:~/the/projects$
```

دقت و زمان اجرا در عکس آخر قابل مشاهده‌اند.
اگر K برابر ۱۰ باشد:

```
s107@sumi:~/the/project$ ./knn 10 10
Device Name: GeForce GTX 1080
ground truths read
Euclidean      N is 0, gth is: 932085          and prediction is: 932085

Query num = 0

Euclidean      N is 1, gth is: 934876          and prediction is: 934876
Euclidean      N is 2, gth is: 561813          and prediction is: 561813
Euclidean      N is 3, gth is: 708177          and prediction is: 708177
Euclidean      N is 4, gth is: 706771          and prediction is: 706771
Euclidean      N is 5, gth is: 695756          and prediction is: 695756
Euclidean      N is 6, gth is: 435345          and prediction is: 435345
Euclidean      N is 7, gth is: 701258          and prediction is: 701258
Euclidean      N is 8, gth is: 455537          and prediction is: 455537
Euclidean      N is 9, gth is: 872728          and prediction is: 872728
Euclidean      N is 0, gth is: 413247          and prediction is: 413247

Query num = 1

Euclidean      N is 1, gth is: 413071          and prediction is: 413071
Euclidean      N is 2, gth is: 706838          and prediction is: 706838
Euclidean      N is 3, gth is: 880592          and prediction is: 880592
Euclidean      N is 4, gth is: 249062          and prediction is: 249062
Euclidean      N is 5, gth is: 400194          and prediction is: 400194
Euclidean      N is 6, gth is: 942339          and prediction is: 942339
Euclidean      N is 7, gth is: 880462          and prediction is: 880462
Euclidean      N is 8, gth is: 987636          and prediction is: 987636
Euclidean      N is 9, gth is: 941776          and prediction is: 941776
Euclidean      N is 0, gth is: 669835          and prediction is: 669835

Query num = 2

Euclidean      N is 1, gth is: 408764          and prediction is: 408764
Euclidean      N is 2, gth is: 408462          and prediction is: 408462
Euclidean      N is 3, gth is: 408855          and prediction is: 408855
Euclidean      N is 4, gth is: 551661          and prediction is: 551661
Euclidean      N is 5, gth is: 861882          and prediction is: 861882
Euclidean      N is 6, gth is: 406273          and prediction is: 406273
Euclidean      N is 7, gth is: 406324          and prediction is: 406324
Euclidean      N is 8, gth is: 551743          and prediction is: 551743
Euclidean      N is 9, gth is: 861530          and prediction is: 861530
Euclidean      N is 0, gth is: 970797          and prediction is: 970797

Query num = 0

Euclidean      N is 1, gth is: 125539          and prediction is: 125539
Euclidean      N is 2, gth is: 48044   and prediction is: 48044
Euclidean      N is 3, gth is: 191115          and prediction is: 191115
Euclidean      N is 4, gth is: 889039          and prediction is: 889039
Euclidean      N is 5, gth is: 546470          and prediction is: 546470
Euclidean      N is 6, gth is: 123985          and prediction is: 123985
Euclidean      N is 7, gth is: 546496          and prediction is: 546496
Euclidean      N is 8, gth is: 207768          and prediction is: 207768
```

```
Query num = 1

Manhattan      N is 1, gth is: NULL      and prediction is: 561813
Manhattan      N is 2, gth is: NULL      and prediction is: 932085
Manhattan      N is 3, gth is: NULL      and prediction is: 893601
Manhattan      N is 4, gth is: NULL      and prediction is: 695756
Manhattan      N is 5, gth is: NULL      and prediction is: 701258
Manhattan      N is 6, gth is: NULL      and prediction is: 36538
Manhattan      N is 7, gth is: NULL      and prediction is: 706771
Manhattan      N is 8, gth is: NULL      and prediction is: 225116
Manhattan      N is 9, gth is: NULL      and prediction is: 722642
Manhattan      N is 0, gth is: NULL      and prediction is: 413247

Query num = 2

Manhattan      N is 1, gth is: NULL      and prediction is: 413071
Manhattan      N is 2, gth is: NULL      and prediction is: 880592
Manhattan      N is 3, gth is: NULL      and prediction is: 880462
Manhattan      N is 4, gth is: NULL      and prediction is: 517923
Manhattan      N is 5, gth is: NULL      and prediction is: 706838
Manhattan      N is 6, gth is: NULL      and prediction is: 249062
Manhattan      N is 7, gth is: NULL      and prediction is: 961942
Manhattan      N is 8, gth is: NULL      and prediction is: 400194
Manhattan      N is 9, gth is: NULL      and prediction is: 987636
Manhattan      N is 0, gth is: NULL      and prediction is: 408764

Query num = 0

Manhattan      N is 1, gth is: NULL      and prediction is: 669835
Manhattan      N is 2, gth is: NULL      and prediction is: 408855
Manhattan      N is 3, gth is: NULL      and prediction is: 104247
Manhattan      N is 4, gth is: NULL      and prediction is: 408462
Manhattan      N is 5, gth is: NULL      and prediction is: 900997
Manhattan      N is 6, gth is: NULL      and prediction is: 534822
Manhattan      N is 7, gth is: NULL      and prediction is: 551661
Manhattan      N is 8, gth is: NULL      and prediction is: 150588
Manhattan      N is 9, gth is: NULL      and prediction is: 609782
Manhattan      N is 0, gth is: NULL      and prediction is: 889039

Query num = 1

Manhattan      N is 1, gth is: NULL      and prediction is: 970797
Manhattan      N is 2, gth is: NULL      and prediction is: 546470
Manhattan      N is 3, gth is: NULL      and prediction is: 191115
Manhattan      N is 4, gth is: NULL      and prediction is: 546767
Manhattan      N is 5, gth is: NULL      and prediction is: 546496
Manhattan      N is 6, gth is: NULL      and prediction is: 125539
Manhattan      N is 7, gth is: NULL      and prediction is: 190621
Manhattan      N is 8, gth is: NULL      and prediction is: 546766
Manhattan      N is 9, gth is: NULL      and prediction is: 190692
Manhattan      N is 0, gth is: NULL      and prediction is: 175336
```

```

Cosine  N is 0, gth is: NULL      and prediction is: 814846

Query num = 2

Cosine  N is 1, gth is: NULL      and prediction is: 218777
Cosine  N is 2, gth is: NULL      and prediction is: 478422
Cosine  N is 3, gth is: NULL      and prediction is: 146642
Cosine  N is 4, gth is: NULL      and prediction is: 43037
Cosine  N is 5, gth is: NULL      and prediction is: 559052
Cosine  N is 6, gth is: NULL      and prediction is: 43395
Cosine  N is 7, gth is: NULL      and prediction is: 43196
Cosine  N is 8, gth is: NULL      and prediction is: 217725
Cosine  N is 9, gth is: NULL      and prediction is: 559294
Cosine  N is 0, gth is: NULL      and prediction is: 443597

Query num = 0

Cosine  N is 1, gth is: NULL      and prediction is: 443442
Cosine  N is 2, gth is: NULL      and prediction is: 443417
Cosine  N is 3, gth is: NULL      and prediction is: 443425
Cosine  N is 4, gth is: NULL      and prediction is: 444021
Cosine  N is 5, gth is: NULL      and prediction is: 444254
Cosine  N is 6, gth is: NULL      and prediction is: 443473
Cosine  N is 7, gth is: NULL      and prediction is: 444363
Cosine  N is 8, gth is: NULL      and prediction is: 444542
Cosine  N is 9, gth is: NULL      and prediction is: 443486
Cosine  N is 0, gth is: NULL      and prediction is: 986614

Query num = 1

Cosine  N is 1, gth is: NULL      and prediction is: 191391
Cosine  N is 2, gth is: NULL      and prediction is: 855890
Cosine  N is 3, gth is: NULL      and prediction is: 850681
Cosine  N is 4, gth is: NULL      and prediction is: 856183
Cosine  N is 5, gth is: NULL      and prediction is: 976474
Cosine  N is 6, gth is: NULL      and prediction is: 51884
Cosine  N is 7, gth is: NULL      and prediction is: 190555
Cosine  N is 8, gth is: NULL      and prediction is: 135183
Cosine  N is 9, gth is: NULL      and prediction is: 969146
Cosine  N is 0, gth is: NULL      and prediction is: 776353

Query num = 2

Cosine  N is 1, gth is: NULL      and prediction is: 776673
Cosine  N is 2, gth is: NULL      and prediction is: 969145
Cosine  N is 3, gth is: NULL      and prediction is: 677423
Cosine  N is 4, gth is: NULL      and prediction is: 824142
Cosine  N is 5, gth is: NULL      and prediction is: 411793
Cosine  N is 6, gth is: NULL      and prediction is: 922585
Cosine  N is 7, gth is: NULL      and prediction is: 960691
Cosine  N is 8, gth is: NULL      and prediction is: 997057
Cosine  N is 9, gth is: NULL      and prediction is: 348715

```

k=10 n=10 GPU=1542.92 ms GPU-Kernels=1304.65 ms accuracy=1.000000

دقت و زمان اجرا در عکس آخر قابل مشاهده‌اند.

اگر K برابر ۱۰۰ باشد:

```
Euclidean    N is 81, gth is: 223504      and prediction is: 223504
Euclidean    N is 82, gth is: 881007      and prediction is: 881007
Euclidean    N is 83, gth is: 210866      and prediction is: 210866
Euclidean    N is 84, gth is: 112891      and prediction is: 112891
Euclidean    N is 85, gth is: 918545      and prediction is: 918545
Euclidean    N is 86, gth is: 629121      and prediction is: 629121
Euclidean    N is 87, gth is: 886772      and prediction is: 886772
Euclidean    N is 88, gth is: 947390      and prediction is: 947390
Euclidean    N is 89, gth is: 844535      and prediction is: 844535
Euclidean    N is 90, gth is: 223613      and prediction is: 223613
Euclidean    N is 91, gth is: 216548      and prediction is: 216548
Euclidean    N is 92, gth is: 751147      and prediction is: 751147
Euclidean    N is 93, gth is: 864289      and prediction is: 864289
Euclidean    N is 94, gth is: 1682      and prediction is: 1682
Euclidean    N is 95, gth is: 142347      and prediction is: 142347
Euclidean    N is 96, gth is: 988960      and prediction is: 988960
Euclidean    N is 97, gth is: 8581      and prediction is: 8581
Euclidean    N is 98, gth is: 629643      and prediction is: 629643
Euclidean    N is 99, gth is: 726241      and prediction is: 726241
Euclidean    N is 0, gth is: 906750      and prediction is: 906750
```

```
Query num = 1
```

```
Euclidean    N is 1, gth is: 618842      and prediction is: 618842
Euclidean    N is 2, gth is: 807599      and prediction is: 807599
Euclidean    N is 3, gth is: 569178      and prediction is: 569178
Euclidean    N is 4, gth is: 107468      and prediction is: 107468
Euclidean    N is 5, gth is: 207868      and prediction is: 207868
Euclidean    N is 6, gth is: 476215      and prediction is: 476215
Euclidean    N is 7, gth is: 762217      and prediction is: 762217
Euclidean    N is 8, gth is: 762237      and prediction is: 762237
Euclidean    N is 9, gth is: 5447      and prediction is: 5447
Euclidean    N is 10, gth is: 112942      and prediction is: 112942
Euclidean    N is 11, gth is: 543133      and prediction is: 543133
Euclidean    N is 12, gth is: 5868      and prediction is: 5868
Euclidean    N is 13, gth is: 108097      and prediction is: 108097
Euclidean    N is 14, gth is: 51349      and prediction is: 51349
Euclidean    N is 15, gth is: 255657      and prediction is: 255657
Euclidean    N is 16, gth is: 325150      and prediction is: 325150
Euclidean    N is 17, gth is: 479784      and prediction is: 479784
Euclidean    N is 18, gth is: 258846      and prediction is: 258846
Euclidean    N is 19, gth is: 235792      and prediction is: 235792
Euclidean    N is 20, gth is: 37728      and prediction is: 37728
Euclidean    N is 21, gth is: 994709      and prediction is: 994709
Euclidean    N is 22, gth is: 570259      and prediction is: 570259
Euclidean    N is 23, gth is: 997837      and prediction is: 997837
Euclidean    N is 24, gth is: 280624      and prediction is: 280624
Euclidean    N is 25, gth is: 259387      and prediction is: 259387
Euclidean    N is 26, gth is: 376741      and prediction is: 376741
Euclidean    N is 27, gth is: 55254      and prediction is: 55254
Euclidean    N is 28, gth is: 695955      and prediction is: 695955
Euclidean    N is 29, gth is: 995722      and prediction is: 995722
Euclidean    N is 30, gth is: 50564      and prediction is: 50564
Euclidean    N is 31, gth is: 124488      and prediction is: 124488
Euclidean    N is 32, gth is: 807699      and prediction is: 807699
Euclidean    N is 33, gth is: 258018      and prediction is: 258018
Euclidean    N is 34, gth is: 476618      and prediction is: 476618
Euclidean    N is 35, gth is: 558460      and prediction is: 558460
Euclidean    N is 36, gth is: 344109      and prediction is: 344109
Euclidean    N is 37, gth is: 890779      and prediction is: 890779
```

Query num = 0

```
Manhattan      N is 1, gth is: NULL      and prediction is: 669835
Manhattan      N is 2, gth is: NULL      and prediction is: 408855
Manhattan      N is 3, gth is: NULL      and prediction is: 104247
Manhattan      N is 4, gth is: NULL      and prediction is: 408462
Manhattan      N is 5, gth is: NULL      and prediction is: 900097
Manhattan      N is 6, gth is: NULL      and prediction is: 534822
Manhattan      N is 7, gth is: NULL      and prediction is: 551661
Manhattan      N is 8, gth is: NULL      and prediction is: 150588
Manhattan      N is 9, gth is: NULL      and prediction is: 609782
Manhattan      N is 10, gth is: NULL     and prediction is: 402106
Manhattan      N is 11, gth is: NULL     and prediction is: 230295
Manhattan      N is 12, gth is: NULL     and prediction is: 534769
Manhattan      N is 13, gth is: NULL     and prediction is: 869379
Manhattan      N is 14, gth is: NULL     and prediction is: 823095
Manhattan      N is 15, gth is: NULL     and prediction is: 104121
Manhattan      N is 16, gth is: NULL     and prediction is: 239766
Manhattan      N is 17, gth is: NULL     and prediction is: 861882
Manhattan      N is 18, gth is: NULL     and prediction is: 929594
Manhattan      N is 19, gth is: NULL     and prediction is: 529249
Manhattan      N is 20, gth is: NULL     and prediction is: 861530
Manhattan      N is 21, gth is: NULL     and prediction is: 937126
Manhattan      N is 22, gth is: NULL     and prediction is: 311367
Manhattan      N is 23, gth is: NULL     and prediction is: 406273
Manhattan      N is 24, gth is: NULL     and prediction is: 905339
Manhattan      N is 25, gth is: NULL     and prediction is: 562123
Manhattan      N is 26, gth is: NULL     and prediction is: 551743
Manhattan      N is 27, gth is: NULL     and prediction is: 306132
Manhattan      N is 28, gth is: NULL     and prediction is: 606268
Manhattan      N is 29, gth is: NULL     and prediction is: 605633
Manhattan      N is 30, gth is: NULL     and prediction is: 406324
Manhattan      N is 31, gth is: NULL     and prediction is: 971815
Manhattan      N is 32, gth is: NULL     and prediction is: 562110
Manhattan      N is 33, gth is: NULL     and prediction is: 467244
Manhattan      N is 34, gth is: NULL     and prediction is: 217709
Manhattan      N is 35, gth is: NULL     and prediction is: 55587
Manhattan      N is 36, gth is: NULL     and prediction is: 534767
Manhattan      N is 37, gth is: NULL     and prediction is: 527380
Manhattan      N is 38, gth is: NULL     and prediction is: 334262
Manhattan      N is 39, gth is: NULL     and prediction is: 401804
Manhattan      N is 40, gth is: NULL     and prediction is: 57338
Manhattan      N is 41, gth is: NULL     and prediction is: 709211
Manhattan      N is 42, gth is: NULL     and prediction is: 744548
Manhattan      N is 43, gth is: NULL     and prediction is: 953237
Manhattan      N is 44, gth is: NULL     and prediction is: 418312
Manhattan      N is 45, gth is: NULL     and prediction is: 743053
Manhattan      N is 46, gth is: NULL     and prediction is: 742644
Manhattan      N is 47, gth is: NULL     and prediction is: 893716
Manhattan      N is 48, gth is: NULL     and prediction is: 502409
Manhattan      N is 49, gth is: NULL     and prediction is: 826055
Manhattan      N is 50, gth is: NULL     and prediction is: 115312
Manhattan      N is 51, gth is: NULL     and prediction is: 716188
Manhattan      N is 52, gth is: NULL     and prediction is: 823487
Manhattan      N is 53, gth is: NULL     and prediction is: 732294
Manhattan      N is 54, gth is: NULL     and prediction is: 983625
Manhattan      N is 55, gth is: NULL     and prediction is: 227542
Manhattan      N is 56, gth is: NULL     and prediction is: 937903
Manhattan      N is 57, gth is: NULL     and prediction is: 238297
```

```
Cosine N is 40, gth is: NULL and prediction is: 969195
Cosine N is 41, gth is: NULL and prediction is: 824016
Cosine N is 42, gth is: NULL and prediction is: 566340
Cosine N is 43, gth is: NULL and prediction is: 443513
Cosine N is 44, gth is: NULL and prediction is: 644422
Cosine N is 45, gth is: NULL and prediction is: 443810
Cosine N is 46, gth is: NULL and prediction is: 544253
Cosine N is 47, gth is: NULL and prediction is: 771080
Cosine N is 48, gth is: NULL and prediction is: 867743
Cosine N is 49, gth is: NULL and prediction is: 669946
Cosine N is 50, gth is: NULL and prediction is: 960710
Cosine N is 51, gth is: NULL and prediction is: 443320
Cosine N is 52, gth is: NULL and prediction is: 807831
Cosine N is 53, gth is: NULL and prediction is: 681907
Cosine N is 54, gth is: NULL and prediction is: 443390
Cosine N is 55, gth is: NULL and prediction is: 39415
Cosine N is 56, gth is: NULL and prediction is: 375823
Cosine N is 57, gth is: NULL and prediction is: 56519
Cosine N is 58, gth is: NULL and prediction is: 195217
Cosine N is 59, gth is: NULL and prediction is: 274033
Cosine N is 60, gth is: NULL and prediction is: 996944
Cosine N is 61, gth is: NULL and prediction is: 807558
Cosine N is 62, gth is: NULL and prediction is: 166289
Cosine N is 63, gth is: NULL and prediction is: 996911
Cosine N is 64, gth is: NULL and prediction is: 976103
Cosine N is 65, gth is: NULL and prediction is: 444953
Cosine N is 66, gth is: NULL and prediction is: 36544
Cosine N is 67, gth is: NULL and prediction is: 969106
Cosine N is 68, gth is: NULL and prediction is: 411685
Cosine N is 69, gth is: NULL and prediction is: 545926
Cosine N is 70, gth is: NULL and prediction is: 444573
Cosine N is 71, gth is: NULL and prediction is: 996909
Cosine N is 72, gth is: NULL and prediction is: 824143
Cosine N is 73, gth is: NULL and prediction is: 851241
Cosine N is 74, gth is: NULL and prediction is: 54648
Cosine N is 75, gth is: NULL and prediction is: 39458
Cosine N is 76, gth is: NULL and prediction is: 443374
Cosine N is 77, gth is: NULL and prediction is: 217920
Cosine N is 78, gth is: NULL and prediction is: 546767
Cosine N is 79, gth is: NULL and prediction is: 919951
Cosine N is 80, gth is: NULL and prediction is: 855550
Cosine N is 81, gth is: NULL and prediction is: 443373
Cosine N is 82, gth is: NULL and prediction is: 846404
Cosine N is 83, gth is: NULL and prediction is: 443393
Cosine N is 84, gth is: NULL and prediction is: 371235
Cosine N is 85, gth is: NULL and prediction is: 443338
Cosine N is 86, gth is: NULL and prediction is: 445007
Cosine N is 87, gth is: NULL and prediction is: 56982
Cosine N is 88, gth is: NULL and prediction is: 918586
Cosine N is 89, gth is: NULL and prediction is: 955990
Cosine N is 90, gth is: NULL and prediction is: 445115
Cosine N is 91, gth is: NULL and prediction is: 652884
Cosine N is 92, gth is: NULL and prediction is: 952227
Cosine N is 93, gth is: NULL and prediction is: 512134
Cosine N is 94, gth is: NULL and prediction is: 960647
Cosine N is 95, gth is: NULL and prediction is: 39573
Cosine N is 96, gth is: NULL and prediction is: 38481
Cosine N is 97, gth is: NULL and prediction is: 919420
Cosine N is 98, gth is: NULL and prediction is: 922313
Cosine N is 99, gth is: NULL and prediction is: 944608
```

k=100 n=10 GPU=1554.18 ms GPU-Kernels=1309.53 ms accuracy=1.000000

دقت و زمان اجرا در عکس آخر قابل مشاهده‌اند.