1. Yu-CNN
   1. 메인 메모리 기반 그리드 색인 구조, 데이터 색인과 질의 색인 제안
   2. 데이터 색인 기법: 오버홀 알고리즘, 점진적 알고리즘 -> 질의의 결과를 주기적으로 재 검토
      1. 질의의 점을 기준으로 k개의 데이터를 포함시킬 때까지 사각형 R을 확장
      2. R에 포함된 데이터와 질의 점 사이의 거리를 d라고 할 때 를 한 변의 길이로 하는 사각형 내의 모든 데이터를 검색
   3. 오버홀 알고리즘: 매 주기마다 모든 데이터의 위치를 기반으로 색인 구조 재구축 후 질의 재 수행
   4. 점진적 알고리즘: 질의 결과에 포함되어 있던 데이터 중 d값이 가장 큰 를 활용하여 인 사각형을 검색 범위로 설정하여 재검색 수행
   5. 질의 색인 기법: 질의의 수가 적을 때만 좋은 성능
   6. 이동 데이터의 위치가 변할 때마다 서버에 보고
   7. 이동 데이터&질의의 수가 증가 시 서버의 병목현상 발생
2. SEA-CNN
   1. 디스크 기반 그리드 색인 구조
   2. 모니터링 영역 개념 활용 -> 초기 결과에 대한 연산보다 점진적인 결과 갱신에 초점
   3. 연속 이동 질의 처리
   4. 일정 주기 마다 다음 단계를 순차적으로 수행(모니터링 범위: 질의 점을 중심으로 하고 질의 점과 k번째 데이터 사이 거리를 반지름으로 하는 원)
      1. 질의 q의 이전 결과에 포함된 데이터가 모니터링 영역 내에서 이동한 경우 or 모니터링 영역 외부 데이터가 영역 내부로 들어온 경우: 재 수행 검색 범위는 모니터링 범위다.
      2. 질의 q의 이전 결과에 포함된 데이터가 모니터링 영역을 벗어난 경우: 재 수행 검색 범위의 반지름은 질의 점과 가장 거리가 먼 데이터 사이의 거리 이다.
      3. 질의 점의 위치가 변했을 경우: 중심은 질의 점, 반지름은 와 질의 점의 변위의 합으로 하는 원을 재 수행 검색 범위로 한다.
   5. 단계 수행 완료 후 질의 연산 수행 및 모니터링 영역을 재설정한다.
   6. 이동 데이터의 위치가 변할 때마다 서버에 보고
   7. 이동 데이터&질의의 수가 증가 시 서버의 병목현상 발생
   8. 이동 데이터에 대한 연속 이동 질의 처리 비용이 큼
3. CPM
   1. 메인 메모리 기반 그리드 색인 구조
   2. 질의 위치를 포함하는 셀을 중심으로 Up, Down, Right, Left의 4가지 방향으로 전체 도메인의 셀들을 분할
   3. 사각형 rect는 방향과 레벨(질의 위치를 포함하는 셀과 rect 사이에 존재하는 사각형 수)로 정의
   4. 힙 초기화 후 다음 연산 수행
      1. 힙에서 삭제한 엔트리가 셀일 경우: 셀 안의 데이터 중 현재 질의 결과에 포함된 데이터보다 가까운 데이터가 존재하면 결과 갱신
      2. 힙에서 삭제한 엔트리가 사각형일 경우: 해당 사각형을 구성하는 셀들과 다음 레벨의 사각형을 힙에 삽입
   5. 질의 점과 가장 먼 데이터 사이의 거리를 반지름으로 하는 원을 모니터링 영역으로 설정
   6. 모니터링 영역 내로 이동한 데이터 수 < 모니터링 영역 밖으로 이동한 데이터 수: 질의 결과 갱신 (모니터링 영역 내로 유입되는 데이터 수가 클 경우 연산이 복잡해질 가능성 有)
   7. 다양한 자료구조 사용으로 인한 메모리 사용이 큼
4. iSEE
   1. 메인 메모리 기반 그리드 색인 구조
   2. VOB: 질의 점에서부터 그리드 셀까지의 최소 거리가 유사한 셀들끼리 레벨 설정하고 그룹화
   3. 그룹별로 힙을 초기화한 후 CPM과 유사하게 연산을 반복 수행
   4. 다양한 자료구조 사용으로 인한 메모리 사용이 큼
5. disMKNN
   1. 분산화된 환경
   2. 하나의 기지국이 수용할 수 있는 범위를 하나의 셀로 정의
   3. 초기 결과 검색 연산
      1. 질의가 속한 셀 및 인접 셀을 조사하여 k개의 임시 결과 값을 찾는다.
      2. 가장 먼 데이터와의 거리를 반지름으로 하는 원과 겹치는 셀들을 조사하여 k개의 데이터를 검색
      3. k번째 데이터와 질의 사의 거리를 반지름으로 하는 원과 겹치는 셀(IC)에 포함된 데이터들에게 질의 생성자와 k번째 데이터의 위치 및 속도 벡터를 전송
   4. 지속적 갱신 연산
      1. IC 내의 데이터는 전송된 정소를 바탕으로 자신이 질의 결과에 포함되는 시간을 계산하여 해당 시간에 서버에게 자신의 위치 및 속도 벡터를 전송
      2. 서버가 질의 결과를 갱신 후 IC에 포함된 데이터에게 새로운 질의 결과를 전송
   5. 셀의 크기가 시스템 성능을 좌우
   6. 셀끼리 겹치는 영역 발생