**📘 Graph-Based Campus Navigation System (Maths-Only Approach)**

**1. Graph Theory (Discrete Mathematics)**

* **Vertices (V):** Each location in campus (Library, Hostel, Canteen, Classrooms).
* **Edges (E):** Paths connecting locations.
* **Weights (w):** Distance (meters) or time (minutes).

Mathematical Representation:

G=(V,E)G = (V, E)G=(V,E)

where V={v1,v2,v3,...}V = \{v\_1, v\_2, v\_3, ...\}V={v1​,v2​,v3​,...}, E⊆V×VE \subseteq V \times VE⊆V×V.

* **Adjacency Matrix:**  
  If there are nnn locations, create an n×nn \times nn×n matrix AAA where:

Aij={distance between vi and vjif edge exists∞otherwiseA\_{ij} = \begin{cases} \text{distance between } v\_i \text{ and } v\_j & \text{if edge exists} \\ \infty & \text{otherwise} \end{cases}Aij​={distance between vi​ and vj​∞​if edge existsotherwise​

**2. Linear Algebra (Matrices)**

* Store campus connections in **adjacency matrix**.
* Use **matrix multiplication** to find paths of length 2 or more.
* Example:  
  If AAA is the adjacency matrix, then

(Ak)ij(A^k)\_{ij}(Ak)ij​

gives number of distinct paths of length kkk from viv\_ivi​ to vjv\_jvj​.

So, by **powers of matrices**, we can analyze reachability in the campus.

**3. Shortest Path (Optimization Concept)**

* Problem = **Find minimum distance between two nodes**.
* Using **Dijkstra’s algorithm (Greedy Method):**

d(v)=min⁡{d(u)+w(u,v)}d(v) = \min \{ d(u) + w(u,v) \}d(v)=min{d(u)+w(u,v)}

where d(v)d(v)d(v) is shortest distance to node vvv.

* Using **Floyd–Warshall (Dynamic Programming):**

Dij(k)=min⁡(Dij(k−1),Dik(k−1)+Dkj(k−1))D\_{ij}^{(k)} = \min \left( D\_{ij}^{(k-1)}, D\_{ik}^{(k-1)} + D\_{kj}^{(k-1)} \right)Dij(k)​=min(Dij(k−1)​,Dik(k−1)​+Dkj(k−1)​)

where Dij(k)D\_{ij}^{(k)}Dij(k)​ = shortest path between iii and jjj using first kkk nodes.

**4. Calculus (Optimization & Rates)**

* If weights represent **time depending on speed**,

Time=DistanceSpeed\text{Time} = \frac{\text{Distance}}{\text{Speed}}Time=SpeedDistance​

* If congestion changes speed, then speed can be modeled as a function v(t)v(t)v(t).

T=∫dsv(t)T = \int \frac{ds}{v(t)}T=∫v(t)ds​

(integration gives total travel time).

**5. Probability & Statistics**

* If multiple paths exist, we can assign **probability of congestion** to each.
* Example: Path A has 70% chance of being clear, Path B has 40%.
* Expected Travel Time:

E[T]=∑(pi×ti)E[T] = \sum (p\_i \times t\_i)E[T]=∑(pi​×ti​)

**6. Final Mathematical Model**

* Model campus as **weighted graph** using adjacency matrix (Linear Algebra).
* Apply **Graph Theory algorithms** (Dijkstra/Floyd).
* Use **Calculus** to minimize time considering speed/crowd.
* Use **Probability** for uncertain conditions.