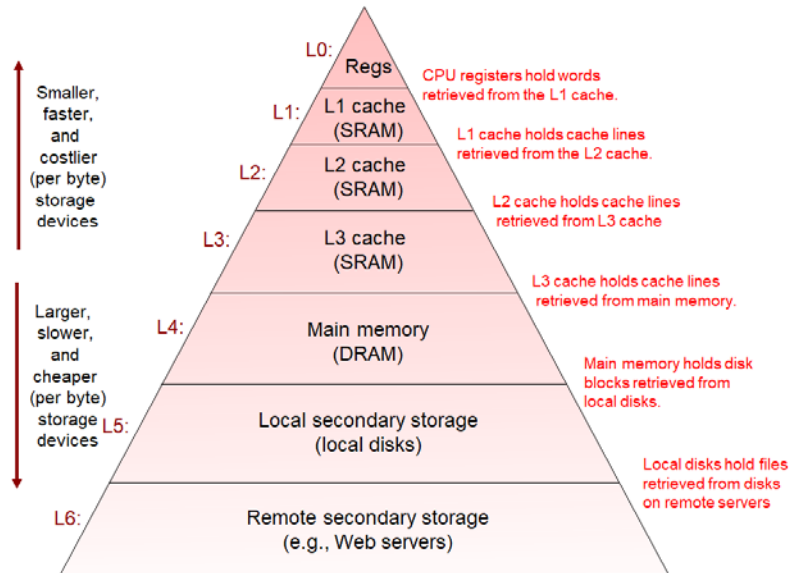


➤ 2nd Project

- Refer to the following figure.



- Guideline
 - You can use any programming languages (e.g., C/C++, Java, or Python) only if they support arrays
 - Design your own memory hierarchy along with 4 layers where each one can be represented by an array. That is, you can suggest four kinds of arrays having different number of elements: 1) L1(1 element), 2) L2(16 elements), 3) L3(256 elements), and 4) L4(4096 elements). Here, the number of elements in each layer can be changed considering the used data.
 - Choose the real-world data to store them to the 4-layer memory hierarchy and collect them. The data needs to be larger than the biggest array (i.e., larger than 4096).
 - Validate the memory hierarchy by storing the data to the memory hierarchy.
 - Apply E-way set associative cache ($E=2$) to L3 layer.
 - Show the cache hit ratio for a given data set based on your own real-world scenario.
- Check list (source codes must include these items; they and their results are needed to be explained in the presentation; the score portion for each item will be varied according to the relative difficulty)**
 - The designed memory hierarchy consists of 4 layers to show the effectiveness of memory hierarchy considering the used real-world data.
 - The real-world data is large enough so that only some of them can be maintained in the memory hierarchy.
 - E-way set associate cache works correctly. This should be shown using an actual example of the real-world data in the presentation.
 - The cache hit ratio is correctly calculated. This should be shown by the presentation based on the real-world scenario on the data.

- **Be aware of cheating! The cheating will be detected based on the submitted source codes.**
- Presentation and submission
 - Presentation: 12 Dec. 2022 at 11:00 AM @ Frontier 511
 - Submit the presentation slide and supplementary materials (including source codes) before the presentation; the results will be verified by them after the presentation
 - Late submissions are not allowed