Anecdotal stories are great to read but I like to make decisions on data. I haven’t seen much data when it comes to which system is better for memorizing numbers or long lists of digits as in a memory competition and I needed to do better at house numbers and phone numbers. So here is my analysis on systems for competition and casual users like me. Please feel free to suggest improvements or other systems.

# Analysis of number peg systems

## Summary

The **3-digit story** **system is best** for competition if several years of practice are put in. A similar 2-digit system is best for casual users producing faster results than any peg system. . A similar 4-digit system has little added benefit and practice becomes burdensome

## The winning systems

The top-ranked 3-digit improvisational story system starts with an encoding system to map digits to word sounds. Popular encodings from digit to word sounds are the Major and the Ben systems (@Ben = Ben Pridmore). Mappings to other data types can be made for 2-digit systems but don’t have the flexibility that sounds give. The word sounds are then visualized as a memory image.

The Ben system has the added benefit for memory competition of reuse for binary numbers and playing cards because of the 16 different consonant sounds. The Major system can be extended into 13 consonant sounds easily for playing cards and 16 takes a little more effort.

### Systems with SVE =< 1.0 and low mastery time

SVE = System visualization efficiency

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rank** | **System** | **Description** | **SVE** | **Mastery time** | **notes** |
| 1 | s-3-\* | story-based 3-digit any data type | 0.7 | 3y | best for competition users |
| 2 | s-2-\* | story-based 3-digit any data type | 1.0 | 3w | best for casual users |
| 3 | mol-3-\* | method of loci-based 3 digit any data type | 1.0 | 3y |  |
| 4 | s-1c2-\* | story-based 3-digit any data type using 1 category | 1.0 | 2y10m |  |

## Discussion

In general, the systems using a narrative system had a greater efficiency seen by the smaller SVE but topped out at three digits for significant incremental gains. The preparation workload between three- and four-digit narrative systems was greatly increased.

@katiek (Katie Kermode) switched from a modified 3-PAO (SVE=2.3) system to a 3-\* (SVE=0.7) type system at the direction of @lociinthesky (Lance Tschirhart) and increased her speed after eight years of using the 3-PAO system in only two months.

I think that the placement of the optimum number memory images in the same location in a method of loci background, is a matter of the Miller number. In other words, the mental workload decreases if you place two rather than three memory images per location. @AlexM (Alex Mullen) switched from three to two memory images and believes that the change was responsible for a spike in his scores. I don’t know his number system but any decrease in visualization workload towards a goal of five to seven will get you the best results.

The 3-digit improvisational story visualizationseems to be best for competition if several years of practice are put in which most competitors do. A similar 4-digit system has little added benefit and the practice of it becomes burdensome. Most people want to develop a competition weight system quickly. A similar 2-digit system is best for casual users producing faster results than any peg system.

One thing I’ve found that helps in developing a narrative traversal approach is for the memory images to contain significant enhancing details with a visual sentence structure using subject, subject enhancements, strong action verb, direct object or items, and location or terrain (SEA-IT). The ability to tie a new image to the last is increased by the details that you can build relevant associations to.

Competitors with an already strong reusable method of loci system have no benefit over forming visual sentences and this would be a personal choice. Casual users do best with less complexity and therefore a story system with a narrative traversal is better to start with.

## System construction

### Data types

Systems and other terms are described by names defined in my **Memory systems summary** and **Glossary of memory terms** found at [my Github repo for memory](https://github.com/doughoff/Memory).

Data types are classified into the five basic SEA-IT types:

* Subject (person, group, organization, role, living thing, fictional character)
* Enhancements for the subject (tools, costume, expression, shape, size, body features)
* Action (a strong verb)
* Inanimate direct objects or items
* Terrain or location.

### Single or multiple data types

A peg system maps an ordered set of data to memory images. The memory image can be a simple data type or multiple unique SEA-IT data types can be built into a more complex visual sentence for easier recall.

The single data type is the first one people learn as in the number pegs associated with the Major system:

|  |  |  |
| --- | --- | --- |
| # | Major system sound | Major system peg |
| 0 | S, Z | saw |
| 1 | D, T, TH | tie |
| 2 | N | Noah |
| 3 | M | ma |
| 4 | R | rye |
| 5 | L | law |
| 6 | CH, J, SH | shoe |
| 7 | C, G, K, Q | cow |
| 8 | F, PH, V | ivy |
| 9 | B, P | bee |

In order for a system that uses multiple data types to work properly, each one of the pegs must be unique in the visual sentence. If you were to use a SEA-IT system using the Major encoding system, it might look like:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Subject** | **Enhancements** | **Action** | **Items** | **Terrain** |
| 5 | Albert Einstein | frizzy white hair | is writing math formulas | on a blackboard with chalk | in a classroom. |

The more complete the visual image become by using more of the data types, the less ability you have to add enhancing details to your image in order to associate them with another memory image in a narrative type of traversal. The less complete the SEA-IT structure is, the more you can use more relevant imagery. For that reason, the SA system is useful in creating stories and can be enhanced at will. But the full SEA-IT visual image is better deposited in a location in a method of loci system. The SAI type system, more commonly called the PAO system is a good compromise for using in a narrative story or with the method of loci because of that.

### Encoding chunk size

The number of items that are encoded to one data type at a time helps speed up the efficiency of the system. For digits they are known as a 1-, 2-, or 3-digit system. 1-digit systems have a problem with creating overused pieces of data. 2-digit systems require the system builder to provide 100 memory images per data type. And 3-digit systems are difficult to prepare for use because of the complexity of conversion and difficult to visualize because of the practice time involved to keep it in practice./.

Different types of rules exist to deal with issues when the encoding chunk size does not evenly fit the system. For instance, using a 3-digit system for a phone number uses one memory image for the area code, another memory image for the prefix and the last four numbers create a problem. If you create a memory image for the first three digits, then the last digit has to be visualized and associated with the other three. What if two digits are left over?

The best way that I’ve seen to handle that is to create a 1-digit system to use with the 2-digit system for leftovers. And for 3-digit systems, you would need complete 1- and 2-digit systems to resolve issues.

### Traversal rules

After the memory image is created, it needs to be associated with something for it to be remembered in order. That is the traversal rule that is a choice of whether to use a narrative story or the method of loci type system. In the narrative story, the last image is used to form an association with the next one and if need be, enhancing details are imagined in order to bring the images to life with hooks to allow for better linking up with the next image. The enhancing visual details are usually the remaining data types not being used by the system in the SEA-IT set of imagery.

The other type of traversal method is based on dropping a memory image in a background at a location that is traversed by any rule that has been defined for that type of method of loci. It could be your typical morning preparation routine in your house or it could be a set of significant places along a walking path. In any case, the memory image is in need of another set of ordered containers so that the memory images can be traversed from start to finish for long strings of digits.

The story style traversal works easily for small groups of memory images up to around five to seven so that you can see it as a scene. And after that a method of loci provides connecting backgrounds for more than seven locations. Some people place two or three memory images in each location of a background and a rule is needed to distinguish the order of those images at the same place. It could be a short story associating the two or it could be a rule of a special trait used for the order such as size or shape.

### Memory image complexity

The number of items that can be visualized into one memory image is limited by the Miller number which is an approximate value of what is capable of being visualized in working memory. The number of data types that are combined in one visual sentence should be restricted to seven give or take two types. I believe that five is an optimum number to work with but with practice, I’m sure people could push the limit up to seven. The system becomes harder to prepare when minor data types are selected, and five data types take care of most of the systems that people use with 2- and 3-digit systems. A 1-digit system would be able to hold more data types in memory easily but would require more research for each subject.

Data types must be associated to each other to be considered one system. If several are used in combination without an association between them, then the systems are not one system.

### Pivot peg type

Data types have a primary association type or a pivot peg type that is used to first arrive at a memory image when mapping digits to the system. Most pivot peg types are subjects and then each data type is related to that subject. The previous example uses the subject of Albert Einstein as the pivot peg type to visualize 55555:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Subject** | **Enhancements** | **Action** | **Items** | **Terrain** |
| 5 | **Albert Einstein** | frizzy white hair | is writing math formulas | on a blackboard with chalk | in a classroom. |

It could easily be altered to use the word sound of five (L) in the Major system as the pivot peg type:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | Albert Einstein | in locks and chains | is listening on a phone | to an owl | in Louisville, KY |

The pivot peg type of the word sound does not lend itself to relevancy of memory images and therefore isn’t as strong as one using a different data type. Using the location as the pivot peg type might produce this set of mapped values:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Subject** | **Enhancements** | **Action** | **Items** | **Terrain** |
| 5 | Bill Shoemaker | in a jockey’s jersey | is riding on the back of | a saddled horse | **on a racetrack in Louisville, KY** |

Some people do not use a pivot peg type because it makes the system easier to compile by not restricting the choices. This seems especially true when using categories that have no associations to the word sound because it’s a personal preference to put their favorite category first or relate them by various rules that are not consistent. A category for the Major system word sound of eight (f,v) might be physicist but then someone might know pool players more and choose to associate that with the eight-ball.

### Index peg systems

If a pivot type is associated with a category that it depends on and is not part of the memory image, it is a separate peg system that indexes this peg system. In our last example, Bill Shoemaker is a horse jockey and the fact that he is in a sport is not relevant to the memory image. It however does tie a prefix of zero to the decoded numbers since sports begins with the Major system word sound of s or z.

Index peg systems can also be indexed which could be called a meta-index system, a super-index, or a two-level index peg system. When used with a SEA-IT set of data, an index peg creates a compound peg system.

The number of SEA-IT visual sentences is expanded from the number of items in the system, ten for a 1-digit, 100 for a 2-digit system, by however many categories you choose up to ten. Theoretically, a 3-digit system tied to a 2-level category index system would produce 1000 \* 10 \* 10 = 100,000 different memory images.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 |  | 8 |  |  | 5 | 5 | 5 | 5 | 5 |
| **#** | **Category 1** | **#** | **Category 2** |  | **#** | **Subject** | **Enhancements** | **Action** | **Items** | **Terrain** |
| 0 | Sciences | 8 | Physicist |  | 5 | Albert Einstein | frizzy white hair | is writing math formulas | on a blackboard with chalk | in a classroom. |

## The analysis method

Each system was broken down into two main process categories of preparation and visualization. The competition environment uses simple data types that need no strategies or familiarization except in the case of people’s names where practicing with common names helps. Each process defines a quantifiable amount of mental work as either encodings or associations.

**Preparation workload** involves the development or acquiring and practice of a memory system. The metric used here is the number of visual sentence component pegs (P, A, O, L) to be prepared for the system (CPs). This is the number of digits used in the system multiplied by the number of component peg values needed. Visual sentence component pegs directly relate to the amount of time needed to become proficient through practice and a conversion of one PC per day is used for casual users and is meant for comparison purposes only. The more practice that competitors put in will reduce the time needed to master the system. A lower number is better.

**Visualization workload** is measured by the units of how many associations are made per memory image so that it can be used with another memory image. I assumed that most system users were fluent in encoding digits to sounds and that what mattered was the association of sounds to memory image rather than having any issue with decreased efficiency due to being unable to encode. Each sound association maps a digit or digits to a visualized memory image.

A complex memory image, or visual sentence, is created by adding enhancing details so that it becomes more memorable. Those data type details can include subject, subject enhancements, strong action verb, direct object or items, and location or terrain (SEA-IT). A visual sentence can be described as connected set of those details such as a subject-verb-object in a location. This is a direct relationship with the PAO (Person-Action-Object) system which has been improved by adding a location (PAIL).

The metrics used for measurement per memory image were

1. how many image components were recalled (**MI components**)
2. how many associations were made in the creation of the visual sentence, usually one less than the MI components (**sentence links**) and
3. how many associations need to be made to store the memory image with the backing traversal order in the memory system (**system links**).

Because the mind can store seven plus or minus two items at a time easily in working memory, the Miller number, a goal of seven or less is desired for visualization workload.

That number is then divided by the number of digits being encoded to get a KPI (key performance indicator) for system visualization efficiency (SVE). The lower the score the better.

The other KPI of visual sentence component pegs (CPs) should make systems easily comparable for either casual or competitive users by a simple approximate conversion of one day per peg mastered. Of course, more practice will lower mastery time.

## System definitions

92 total systems. Not all of the permutations of the data types were used but only the ones that seemed to be the logical progression of adding enhancement to the visual sentence moving from any kind of subject or object to a subject and an action, to adding an item or object, to adding a location, and lastly adding enhancing details to the subject. Category based systems were assumed to have leftover systems built in (a 1-digit system in a 2-digit system). Every category is assumed to use the full ten types.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **traversal type** | **# of categories** | **encoding chunk size** | **# of data types** | **abbreviation** | **common names** |
| story | 0 | 1 | 1 | s-1-\* | number peg story |
| method of loci | 0 | 1 | 1 | mol-1-\* |  |
| story | 0 | 1 | 2 | s-1-SA |  |
| method of loci | 0 | 1 | 2 | mol-1-SA |  |
| story | 0 | 1 | 3 | s-1-SAI |  |
| method of loci | 0 | 1 | 3 | mol-1-SAI | 1-digit PAO |
| story | 0 | 1 | 4 | s-1-SAIT |  |
| method of loci | 0 | 1 | 4 | mol-1-SAIT |  |
| story | 0 | 1 | 5 | s-1-SEA-IT |  |
| method of loci | 0 | 1 | 5 | mol-1- SEA-IT |  |
| story | 0 | 2 | 1 | s-2-\* |  |
| method of loci | 0 | 2 | 1 | mol-2-\* |  |
| story | 0 | 2 | 2 | s-2-SA |  |
| method of loci | 0 | 2 | 2 | mol-2-SA |  |
| story | 0 | 2 | 3 | s-2-SAI |  |
| method of loci | 0 | 2 | 3 | mol-2-SAI | 2-digit PAO |
| story | 0 | 2 | 4 | s-2-SAIT |  |
| method of loci | 0 | 2 | 4 | mol-2-SAIT |  |
| story | 0 | 2 | 5 | s-2-SEA-IT |  |
| method of loci | 0 | 2 | 5 | mol-2- SEA-IT |  |
| story | 0 | 3 | 1 | s-3-\* | 1000 object |
| method of loci | 0 | 3 | 1 | mol-3-\* | 1000 object |
| story | 0 | 3 | 2 | s-3-SA |  |
| method of loci | 0 | 3 | 2 | mol-3-SA |  |
| story | 0 | 3 | 3 | s-3-SAI |  |
| method of loci | 0 | 3 | 3 | mol-3-SAI | 3-digit PAO |
| story | 0 | 3 | 4 | s-3-SAIT |  |
| method of loci | 0 | 3 | 4 | mol-3-SAIT |  |
| story | 0 | 3 | 5 | s-3-SEA-IT |  |
| method of loci | 0 | 3 | 5 | mol-3- SEA-IT |  |
| story | 0 | 4 | 1 | s-4-\* |  |
| method of loci | 0 | 4 | 1 | mol-4-\* |  |
| story | 1 | 1 | 1 | s-1c1-\* | categorized number peg story |
| method of loci | 1 | 1 | 1 | mol-1c1-\* |  |
| story | 1 | 1 | 2 | s-1c1-SA |  |
| method of loci | 1 | 1 | 2 | mol-1c1-SA |  |
| story | 1 | 1 | 3 | s-1c1-SAI |  |
| method of loci | 1 | 1 | 3 | mol-1c1-SAI | categorized 1-digit PAO |
| story | 1 | 1 | 4 | s-1c1-SAIT |  |
| method of loci | 1 | 1 | 4 | mol-1c1-SAIT |  |
| story | 1 | 1 | 5 | s-1c1-SEA-IT |  |
| method of loci | 1 | 1 | 5 | mol-1c1- SEA-IT |  |
| story | 1 | 2 | 1 | s-1c2-\* |  |
| method of loci | 1 | 2 | 1 | mol-1c2-\* |  |
| story | 1 | 2 | 2 | s-1c2-SA |  |
| method of loci | 1 | 2 | 2 | mol-1c2-SA |  |
| story | 1 | 2 | 3 | s-1c2-SAI |  |
| method of loci | 1 | 2 | 3 | mol-1c2-SAI | categorized 2-digit PAO |
| story | 1 | 2 | 4 | s-1c2-SAIT |  |
| method of loci | 1 | 2 | 4 | mol-1c2-SAIT |  |
| story | 1 | 2 | 5 | s-1c2-SEA-IT |  |
| method of loci | 1 | 2 | 5 | mol-1c2- SEA-IT |  |
| story | 1 | 3 | 1 | s-1c3-\* | categorized 1000 object |
| method of loci | 1 | 3 | 1 | mol-1c3-\* | categorized 1000 object |
| story | 1 | 3 | 2 | s-1c3-SA |  |
| method of loci | 1 | 3 | 2 | mol-1c3-SA |  |
| story | 1 | 3 | 3 | s-1c3-SAI |  |
| method of loci | 1 | 3 | 3 | mol-1c3-SAI | categorized 3-digit PAO |
| story | 1 | 3 | 4 | s-1c3-SAIT |  |
| method of loci | 1 | 3 | 4 | mol-1c3-SAIT |  |
| story | 1 | 3 | 5 | s-1c3-SEA-IT |  |
| method of loci | 1 | 3 | 5 | mol-1c3- SEA-IT |  |
| story | 2 | 1 | 1 | s-2c1-\* | 2 category number peg story |
| method of loci | 2 | 1 | 1 | mol-2c1-\* |  |
| story | 2 | 1 | 2 | s-2c1-SA |  |
| method of loci | 2 | 1 | 2 | mol-2c1-SA |  |
| story | 2 | 1 | 3 | s-2c1-SAI |  |
| method of loci | 2 | 1 | 3 | mol-2c1-SAI | 2 category 1-digit PAO |
| story | 2 | 1 | 4 | s-2c1-SAIT |  |
| method of loci | 2 | 1 | 4 | mol-2c1-SAIT |  |
| story | 2 | 1 | 5 | s-2c1-SEA-IT |  |
| method of loci | 2 | 1 | 5 | mol-2c1- SEA-IT |  |
| story | 2 | 2 | 1 | s-2c2-\* |  |
| method of loci | 2 | 2 | 1 | mol-2c2-\* |  |
| story | 2 | 2 | 2 | s-2c2-SA |  |
| method of loci | 2 | 2 | 2 | mol-2c2-SA |  |
| story | 2 | 2 | 3 | s-2c2-SAI |  |
| method of loci | 2 | 2 | 3 | mol-2c2-SAI | 2 category 2-digit PAO |
| story | 2 | 2 | 4 | s-2c2-SAIT |  |
| method of loci | 2 | 2 | 4 | mol-2c2-SAIT |  |
| story | 2 | 2 | 5 | s-2c2-SEA-IT |  |
| method of loci | 2 | 2 | 5 | mol-2c2- SEA-IT |  |
| story | 2 | 3 | 1 | s-2c3-\* | 2 category 1000 object |
| method of loci | 2 | 3 | 1 | mol-2c3-\* | 2 category 1000 object |
| story | 2 | 3 | 2 | s-2c3-SA |  |
| method of loci | 2 | 3 | 2 | mol-2c3-SA |  |
| story | 2 | 3 | 3 | s-2c3-SAI |  |
| method of loci | 2 | 3 | 3 | mol-2c3-SAI | 2 category 3-digit PAO |
| story | 2 | 3 | 4 | s-2c3-SAIT |  |
| method of loci | 2 | 3 | 4 | mol-2c3-SAIT |  |
| story | 2 | 3 | 5 | s-2c3-SEA-IT |  |
| method of loci | 2 | 3 | 5 | mol-2c3-SEA-IT |  |

## Analysis results

### Preparation workload

CP = visual sentence component pegs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **system peg values** | **peg images combined** | **CP Total** | **App. time to master** |
| s-1-\* | 10 | 1 | 10 | 1w |
| mol-1-\* | 10 | 1 | 10 | 1w |
| s-1-SA | 10 | 2 | 20 | 3w |
| mol-1-SA | 10 | 2 | 20 | 3w |
| s-1-SAI | 10 | 3 | 30 | 1m |
| mol-1-SAI | 10 | 3 | 30 | 1m |
| s-1-SAIT | 10 | 4 | 40 | 1m1w |
| mol-1-SAIT | 10 | 4 | 40 | 1m1w |
| s-1-SEA-IT | 10 | 5 | 50 | 1m3w |
| mol-1- SEA-IT | 10 | 5 | 50 | 1m3w |
| s-2-\* | 100 | 1 | 100 | 3w |
| mol-2-\* | 100 | 1 | 100 | 3w |
| s-2-SA | 100 | 2 | 200 | 6m |
| mol-2-SA | 100 | 2 | 200 | 6m |
| s-2-SAI | 100 | 3 | 300 | 10m |
| mol-2-SAI | 100 | 3 | 300 | 10m |
| s-2-SAIT | 100 | 4 | 400 | 1y1m |
| mol-2-SAIT | 100 | 4 | 400 | 1y1m |
| s-2-SEA-IT | 100 | 5 | 500 | 1y3m |
| mol-2- SEA-IT | 100 | 5 | 500 | 1y3m |
| s-3-\* | 1000 | 1 | 1000 | 3y |
| mol-3-\* | 1000 | 1 | 1000 | 3y |
| s-3-SA | 1000 | 2 | 2000 | 5y |
| mol-3-SA | 1000 | 2 | 2000 | 5y |
| s-3-SAI | 1000 | 3 | 3000 | 8y |
| mol-3-SAI | 1000 | 3 | 3000 | 8y |
| s-3-SAIT | 1000 | 4 | 4000 | 11y |
| mol-3-SAIT | 1000 | 4 | 4000 | 11y |
| s-3-SEA-IT | 1000 | 5 | 5000 | 14y |
| mol-3- SEA-IT | 1000 | 5 | 5000 | 14y |
| s-4-\* | 10,000 | 1 | 10,000 | 28y |
| mol-4-\* | 10,000 | 1 | 10,000 | 28y |
| s-1c1-\* | 10 | 1 | 110 | 4m |
| mol-1c1-\* | 10 | 1 | 110 | 4m |
| s-1c1-SA | 10 | 2 | 210 | 7m |
| mol-1c1-SA | 10 | 2 | 210 | 7m |
| s-1c1-SAI | 10 | 3 | 310 | 10m |
| mol-1c1-SAI | 10 | 3 | 310 | 10m |
| s-1c1-SAIT | 10 | 4 | 410 | 1y2m |
| mol-1c1-SAIT | 10 | 4 | 410 | 1y2m |
| s-1c1-SEA-IT | 10 | 5 | 510 | 1y5m |
| mol-1c1- SEA-IT | 10 | 5 | 510 | 1y5m |
| s-1c2-\* | 100 | 1 | 1010 | 2y10m |
| mol-1c2-\* | 100 | 1 | 1010 | 2y10m |
| s-1c2-SA | 100 | 2 | 2010 | 5y7m |
| mol-1c2-SA | 100 | 2 | 2010 | 5y7m |
| s-1c2-SAI | 100 | 3 | 3010 | 8y |
| mol-1c2-SAI | 100 | 3 | 3010 | 8y |
| s-1c2-SAIT | 100 | 4 | 4010 | 11y |
| mol-1c2-SAIT | 100 | 4 | 4010 | 11y |
| s-1c2-SEA-IT | 100 | 5 | 5010 | 14y |
| mol-1c2- SEA-IT | 100 | 5 | 5010 | 14y |
| s-1c3-\* | 1000 | 1 | 10,010 | 29y |
| mol-1c3-\* | 1000 | 1 | 10,010 | 29y |
| s-1c3-SA | 1000 | 2 | 20,010 | 56y |
| mol-1c3-SA | 1000 | 2 | 20,010 | 56y |
| s-1c3-SAI | 1000 | 3 | 30,010 | 83y |
| mol-1c3-SAI | 1000 | 3 | 30,010 | 83y |
| s-1c3-SAIT | 1000 | 4 | 40,010 | 111y |
| mol-1c3-SAIT | 1000 | 4 | 40,010 | 111y |
| s-1c3-SEA-IT | 1000 | 5 | 50,010 | 139y |
| mol-1c3- SEA-IT | 1000 | 5 | 50,010 | 139y |
| s-2c1-\* | 10 | 1 | 10,110 | 29y |
| mol-2c1-\* | 10 | 1 | 10,110 | 29y |
| s-2c1-SA | 10 | 2 | 20,110 | 56y |
| mol-2c1-SA | 10 | 2 | 20,110 | 56y |
| s-2c1-SAI | 10 | 3 | 30,110 | 83y |
| mol-2c1-SAI | 10 | 3 | 30,110 | 83y |
| s-2c1-SAIT | 10 | 4 | 40,110 | 111y |
| mol-2c1-SAIT | 10 | 4 | 40,110 | 111y |
| s-2c1-SEA-IT | 10 | 5 | 50,110 | 139y |
| mol-2c1- SEA-IT | 10 | 5 | 50,110 | 139y |
| s-2c2-\* | 100 | 1 | 10,110 | 29y |
| mol-2c2-\* | 100 | 1 | 10,110 | 29y |
| s-2c2-SA | 100 | 2 | 20,110 | 56y |
| mol-2c2-SA | 100 | 2 | 20,110 | 56y |
| s-2c2-SAI | 100 | 3 | 30,110 | 83y |
| mol-2c2-SAI | 100 | 3 | 30,110 | 83y |
| s-2c2-SAIT | 100 | 4 | 40,110 | 111y |
| mol-2c2-SAIT | 100 | 4 | 40,110 | 111y |
| s-2c2-SEA-IT | 100 | 5 | 50,110 | 139y |
| mol-2c2- SEA-IT | 100 | 5 | 50,110 | 139y |
| s-2c3-\* | 1000 | 1 | 100,110 | 278y |
| mol-2c3-\* | 1000 | 1 | 100,110 | 278y |
| s-2c3-SA | 1000 | 2 | 200,110 | 556y |
| mol-2c3-SA | 1000 | 2 | 200,110 | 556y |
| s-2c3-SAI | 1000 | 3 | 300,110 | 834y |
| mol-2c3-SAI | 1000 | 3 | 300,110 | 834y |
| s-2c3-SAIT | 1000 | 4 | 400,110 | 1,111y |
| mol-2c3-SAIT | 1000 | 4 | 400,110 | 1,111y |
| s-2c3-SEA-IT | 1000 | 5 | 500,110 | 1,389y |
| mol-2c3-SEA-IT | 1000 | 5 | 500,110 | 1,389y |

### Visualization workload

#### Systems with SVE =< 1.0

Ranked by SVE then total number of memory image components and associations.

SVE = System visualization efficiency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rank** | **System** | **Description** | **Mastery time** | **SVE** |
| 1 | s-4-\* | story-based 4-digit any data type | 28y | 0.5 |
| 2 | s-3-\* | story-based 3-digit any data type | 3y | 0.7 |
| 3 | s-1c3-\* | story-based 3-digit any data type using 1 category | 29y | 0.8 |
| 4 | s-2c3-\* | story-based 3-digit any data type using 2 categories | 278y | 0.8 |
| 5 | s-2-\* | story-based 3-digit any data type | 3w | 1.0 |
| 6 | mol-3-\* | method of loci-based 3 digit any data type | 3y | 1.0 |
| 7 | s-1c2-\* | story-based 3-digit any data type using 1 category | 2y10m | 1.0 |
| 8 | s-2c2-\* | story-based 3-digit any data type using 2 categories | 29y | 1.0 |
| 9 | mol-1c3-\* | method of loci-based 3-digit any data type using 1 category | 29y | 1.0 |
| 10 | mol-2c3-\* | method of loci-based 3-digit any data type using 2 categories | 278y | 1.0 |

#### All systems

MIC = Memory image components

SCA = Sentence component associations (MIC - 1)

SIA = System image associations

SVE = System visualization efficiency

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **System** | **MIC** | **+SCA** | **+SIA** | **=Total  (7 ± 2 max)** | **/Chunks** | **= SVE** |
| s-1-\* | 1 | 0 | 1 | 2 | 1 | 2.0 |
| mol-1-\* | 1 | 0 | 2 | 3 | 1 | 3.0 |
| s-1-SA | 2 | 1 | 1 | 4 | 1 | 4.0 |
| mol-1-SA | 2 | 1 | 2 | 5 | 1 | 5.0 |
| s-1-SAI | 3 | 2 | 1 | 6 | 1 | 6.0 |
| mol-1-SAI | 3 | 2 | 2 | 7 | 1 | 7.0 |
| s-1-SAIT | 4 | 3 | 1 | 8 | 1 | 8.0 |
| mol-1-SAIT | 4 | 3 | 2 | 9 | 1 | 9.0 |
| s-1-SEA-IT | 5 | 4 | 1 | 10 | 1 | 10.0 |
| mol-1- SEA-IT | 5 | 4 | 2 | 11 | 1 | 11.0 |
| s-2-\* | 1 | 0 | 1 | 2 | 2 | 1.0 |
| mol-2-\* | 1 | 0 | 2 | 3 | 2 | 1.5 |
| s-2-SA | 2 | 1 | 1 | 4 | 2 | 2.0 |
| mol-2-SA | 2 | 1 | 2 | 5 | 2 | 2.5 |
| s-2-SAI | 3 | 2 | 1 | 6 | 2 | 3.0 |
| mol-2-SAI | 3 | 2 | 2 | 7 | 2 | 3.5 |
| s-2-SAIT | 4 | 3 | 1 | 8 | 2 | 4.0 |
| mol-2-SAIT | 4 | 3 | 2 | 9 | 2 | 4.5 |
| s-2-SEA-IT | 5 | 4 | 1 | 10 | 2 | 5.0 |
| mol-2- SEA-IT | 5 | 4 | 2 | 11 | 2 | 5.5 |
| s-3-\* | 1 | 0 | 1 | 2 | 3 | 0.7 |
| mol-3-\* | 1 | 0 | 2 | 3 | 3 | 1.0 |
| s-3-SA | 2 | 1 | 1 | 4 | 3 | 1.3 |
| mol-3-SA | 2 | 1 | 2 | 5 | 3 | 1.7 |
| s-3-SAI | 3 | 2 | 1 | 6 | 3 | 2.0 |
| mol-3-SAI | 3 | 2 | 2 | 7 | 3 | 2.3 |
| s-3-SAIT | 4 | 3 | 1 | 8 | 3 | 2.7 |
| mol-3-SAIT | 4 | 3 | 2 | 9 | 3 | 3.0 |
| s-3-SEA-IT | 5 | 4 | 1 | 10 | 3 | 3.3 |
| mol-3- SEA-IT | 5 | 4 | 2 | 11 | 3 | 3.7 |
| s-4-\* | 1 | 0 | 1 | 2 | 4 | 0.5 |
| mol-4-\* | 1 | 0 | 2 | 3 | 4 | 1.3 |
| s-1c1-\* | 1 | 0 | 2 | 3 | 2 | 1.5 |
| mol-1c1-\* | 1 | 0 | 3 | 4 | 2 | 2.0 |
| s-1c1-SA | 2 | 1 | 2 | 5 | 2 | 2.5 |
| mol-1c1-SA | 2 | 1 | 3 | 6 | 2 | 3.0 |
| s-1c1-SAI | 3 | 2 | 2 | 7 | 2 | 3.5 |
| mol-1c1-SAI | 3 | 2 | 3 | 8 | 2 | 4.0 |
| s-1c1-SAIT | 4 | 3 | 2 | 9 | 2 | 4.5 |
| mol-1c1-SAIT | 4 | 3 | 3 | 10 | 2 | 5.0 |
| s-1c1-SEA-IT | 5 | 4 | 2 | 11 | 2 | 5.5 |
| mol-1c1- SEA-IT | 5 | 4 | 3 | 12 | 2 | 6.0 |
| s-1c2-\* | 1 | 0 | 2 | 3 | 3 | 1.0 |
| mol-1c2-\* | 1 | 0 | 3 | 4 | 3 | 1.3 |
| s-1c2-SA | 2 | 1 | 2 | 5 | 3 | 1.7 |
| mol-1c2-SA | 2 | 1 | 3 | 6 | 3 | 2.0 |
| s-1c2-SAI | 3 | 2 | 2 | 7 | 3 | 2.3 |
| mol-1c2-SAI | 3 | 2 | 3 | 8 | 3 | 2.7 |
| s-1c2-SAIT | 4 | 3 | 2 | 9 | 3 | 3.0 |
| mol-1c2-SAIT | 4 | 3 | 3 | 10 | 3 | 3.3 |
| s-1c2-SEA-IT | 5 | 4 | 2 | 11 | 3 | 3.7 |
| mol-1c2- SEA-IT | 5 | 4 | 3 | 12 | 3 | 4.0 |
| s-1c3-\* | 1 | 0 | 2 | 3 | 4 | 0.8 |
| mol-1c3-\* | 1 | 0 | 3 | 4 | 4 | 1.0 |
| s-1c3-SA | 2 | 1 | 2 | 5 | 4 | 1.3 |
| mol-1c3-SA | 2 | 1 | 3 | 6 | 4 | 1.5 |
| s-1c3-SAI | 3 | 2 | 2 | 7 | 4 | 1.8 |
| mol-1c3-SAI | 3 | 2 | 3 | 8 | 4 | 2.0 |
| s-1c3-SAIT | 4 | 3 | 2 | 9 | 4 | 2.3 |
| mol-1c3-SAIT | 4 | 3 | 3 | 10 | 4 | 2.5 |
| s-1c3-SEA-IT | 5 | 4 | 2 | 11 | 4 | 2.8 |
| mol-1c3- SEA-IT | 5 | 4 | 3 | 12 | 4 | 3.0 |
| s-2c1-\* | 1 | 0 | 3 | 4 | 3 | 1.3 |
| mol-2c1-\* | 1 | 0 | 4 | 5 | 3 | 1.7 |
| s-2c1-SA | 2 | 1 | 3 | 6 | 3 | 2.0 |
| mol-2c1-SA | 2 | 1 | 4 | 7 | 3 | 2.3 |
| s-2c1-SAI | 3 | 2 | 3 | 8 | 3 | 2.7 |
| mol-2c1-SAI | 3 | 2 | 4 | 9 | 3 | 3.0 |
| s-2c1-SAIT | 4 | 3 | 3 | 10 | 3 | 3.3 |
| mol-2c1-SAIT | 4 | 3 | 4 | 11 | 3 | 3.7 |
| s-2c1-SEA-IT | 5 | 4 | 3 | 12 | 3 | 4.0 |
| mol-2c1- SEA-IT | 5 | 4 | 4 | 13 | 3 | 4.3 |
| s-2c2-\* | 1 | 0 | 3 | 4 | 4 | 1.0 |
| mol-2c2-\* | 1 | 0 | 4 | 5 | 4 | 1.3 |
| s-2c2-SA | 2 | 1 | 3 | 6 | 4 | 1.5 |
| mol-2c2-SA | 2 | 1 | 4 | 7 | 4 | 1.8 |
| s-2c2-SAI | 3 | 2 | 3 | 8 | 4 | 2.0 |
| mol-2c2-SAI | 3 | 2 | 4 | 9 | 4 | 2.3 |
| s-2c2-SAIT | 4 | 3 | 3 | 10 | 4 | 2.5 |
| mol-2c2-SAIT | 4 | 3 | 4 | 11 | 4 | 2.8 |
| s-2c2-SEA-IT | 5 | 4 | 3 | 12 | 4 | 3.0 |
| mol-2c2- SEA-IT | 5 | 4 | 4 | 13 | 4 | 3.3 |
| s-2c3-\* | 1 | 0 | 3 | 4 | 5 | 0.8 |
| mol-2c3-\* | 1 | 0 | 4 | 5 | 5 | 1.0 |
| s-2c3-SA | 2 | 1 | 3 | 6 | 5 | 1.2 |
| mol-2c3-SA | 2 | 1 | 4 | 7 | 5 | 1.4 |
| s-2c3-SAI | 3 | 2 | 3 | 8 | 5 | 1.6 |
| mol-2c3-SAI | 3 | 2 | 4 | 9 | 5 | 1.8 |
| s-2c3-SAIT | 4 | 3 | 3 | 10 | 5 | 2.0 |
| mol-2c3-SAIT | 4 | 3 | 4 | 11 | 5 | 2.2 |
| s-2c3-SEA-IT | 5 | 4 | 3 | 12 | 5 | 2.4 |
| mol-2c3-SEA-IT | 5 | 4 | 4 | 13 | 5 | 2.6 |